NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

CYBERCIEGE SCENARIO ILLUSTRATING INTEGRITY RISKS TO A MILITARY-LIKE FACILITY

by

Klaus W. Fielk

September 2004

Thesis Co-Advisors:	Cynthia E. Irvine
	Paul C. Clark
Second Reader:	Michael F. Thompson

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2004	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Title (Mix case letters) CyberCIEGE Scenario Illustrating Integrity Risks To A Military-Like Facility			**5. FUNDING NUMBERS**
6. AUTHOR(S) Klaus W. Fielk			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

As the number of computer users continues to grow, attacks on assets stored on computer devices have increased. Despite an increase in computer security awareness, many users and policy makers still do not implement security principles in their daily lives. Ineffective education and the lack of personal experience and tacit understanding might be a main cause. The CyberCIEGE game can be used to convey requisite facts and to generate tacit understanding of general computer security concepts to a broad audience.

This thesis asked if a Scenario Definition File (SDF) for the CyberCIEGE game could be developed to educate and train players in Information Assurance on matters related to information integrity in a networking environment. The primary educational concern is the protection of stored data. Another goal was to test whether the game engine properly simulates real world behavior.

The research concluded that it is possible to create SDFs for the CyberCIEGE game engine to teach specifically about integrity issues. Three specific SDFs were developed for teaching purposes. Several SDFs were developed to demonstrate the game engine's ability to simulate real world behavior for specific, isolated educational goals. These tests led to recommendations to improve the game engine.

14. SUBJECT TERMS Computer Security, Integrity, Software Integrity, Trap door, Social Engineering, MAC Enforcement Mechanism, Educational Goals, Information Assurance, CyberCIEGE, Scenario Definition File	**15. NUMBER OF PAGES** 126
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

CYBERCIEGE SCENARIO ILLUSTRATING INTEGRITY RISKS TO A MILITARY LIKE FACILITY

Klaus W. Fielk
Lieutenant, German Navy
Graduate Engineer University in the field of Aeronautical and Space Technology,
University of the German Armed Forces in Munich, 1994

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2004

Author: Klaus W. Fielk

Approved by: Cynthia E. Irvine
Thesis Co-Advisor

Paul C. Clark
Thesis Co-Advisor

Michael F. Thompson
Second Reader

Peter J. Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

As the number of computer users continues to grow, attacks on assets stored on computer devices have increased. Despite an increase in computer security awareness, many users and policy makers still do not implement security principles in their daily lives. Ineffective education and the lack of personal experience and tacit understanding might be a main cause. The CyberCIEGE game can be used to convey requisite facts and to generate tacit understanding of general computer security concepts to a broad audience.

This thesis asked if a Scenario Definition File (SDF) for the CyberCIEGE game could be developed to educate and train players in Information Assurance on matters related to information integrity on a networking environment. The primary educational concern is the protection of stored data. Another goal was to test whether the game engine properly simulates real world behavior.

The research concluded that it is possible to create SDFs for the CyberCIEGE game engine to teach specifically about integrity issues. Three specific SDFs were developed for teaching purposes. Several SDFs were developed to demonstrate the game engine's ability to simulate real world behavior for specific, isolated educational goals. These tests led to recommendations to improve the game engine.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS

CC	Common Criteria
CDS	Combat Direction System
CO	Comanding Officer
DAC	Discretionary Access Control
DOD	Department Of Defense
DOS	Denial Of Service
EAL	Evaluated Assurance Level
EW	Electronic Warfare
IA	Information Assurance
ICMS	Integrated Monitoring and Control System
IS	Information Systems
IT	Information Technology
MAC	Mandatory Access Control
OS	Operating System
SDF	Scenario Definition File
SK	Security Kernel

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

Many individuals have influenced and assisted me during the development of this thesis. I would like to express my deepest appreciation to some of them now.

I would like to thank my wife, Nina, for your understanding and unwavering support, through many long hours of work at school, and at home.

I would like to express thanks to Dr. Cynthia Irvine, Paul Clark, and Mike Thompson, my advisors. Thank you for your patience and constructive critique.

I would also like to extend a special thank you to Marc Meyer, Robert LaMore, Daniel Warren, and JD Fulp. You have all been good friends and provided me with insight, inspiration and good humor.

I would like to send a very special thanks to Justin Lamorie and Catalina Phippen. Your help and friendship has provided me with a unique experience and a great time at the NPS. Many times, you have helped to ground me, and to regain a productive attitude. I very much appreciate your assistance learning abut American culture and improving my American English writing style.

Finally, I would also like to thank Gary Kreeger, and Jean Brennan for the kind, excellent, and professional support you provide me as a student during my academic career at NPS.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. THESIS STATEMENT

This thesis contributes to ongoing research concerning the project CyberCIEGE, conducted at the Naval Postgraduate School. “The purpose of the CyberCIEGE project is to create an Information Assurance (IA) teaching/learning laboratory.”[Irvine1 2003]

The focus of this research is the development of a CyberCIEGE Scenario Definition File (SDF) intended to simultaneously educate users about security matters concerning integrity issues and to entertain them. Additionally this research develops a test plan to evaluate whether the CyberCIEGE game engine performs as expected when provided with a user-defined SDF, i.e. producing results within an expected range.

The goal of the thesis is to answer two questions. First, can a scenario be developed, such that it is simultaneously a playable game and educational tool, while illustrating integrity issues in a military-like networked environment? Second, is it possible to validate that the CyberCIEGE game engine produces expected results from a specific scenario definition file?

B. THESIS SCOPE & LAYOUT

The scope of the thesis is to create a SDF for the CyberCIEGE game that educates and trains players in Information Assurance on matters related to a networking environment. The specific area of research is the protection of a network environment with respect to integrity. The primary concern is the protection of stored data that forms part of the player’s assets, e.g., a database containing weapon information, with the effect, that failure to preserve its integrity might result in engaging a false target. Taking into account the nondeterministic nature of the game engine, the scenario definition file and variations of it, representing legitimate and likely user choices, are used to test, whether the game engine, given these inputs, performs in a manner consistent with the expected results of the specified SDF. The

impact of this research could have benefits for future DOD training and education requirements in the Information Assurance / Network Security area.

The thesis comprises following chapters:

- Chapter I – Introduction – This chapter provides the thesis statement and describes the scope to the thesis. It gives an overview over the chapters, figures and annexes of the paper.
- Chapter II - Background – This chapter describes the project and its background and illustrates the contribution of this thesis to the overall project. It describes some of the key concepts this paper focuses on.
- Chapter III – Educational Goals – This chapter analyses the game's target group and introduces the reader into the specific educational goals of the SDF created with this thesis.
- Chapter IV – Scenario Description – This chapter introduces the reader to the player's virtual world as modeled by the SDF. It includes a narrative description, the briefing to the player, a description of the users, policies, assets, components, and potential attacks, modeled by the SDF.
- Chapter V - Testing – This chapter describes the test strategy and important test cases indicating the scope, the expected and actual results.
- Chapter VI - Future Work & Conclusion Conclusion – This chapter looks at areas of potential further research, and gives a brief summary of the work accomplished by this thesis.

C. APPENDIX OVERVIEW

The following appendices complete this thesis:

- Appendix A - Scenario Definition Files (SDF). This appendix includes the SDF's of the main test cases described in this paper, and the playable SDFs. For the playable SDFs it includes a version of game play with possible player choices that lead to winning the game, and a second version of game play with bad player choices that lead to losing the game.
- Appendix B – Workspace Files. This appendix includes the workspace files used for the scenarios.

II. BACKGROUND

This chapter describes briefly the motivation for the CyberCIEGE game, the current unsatisfactory situation concerning computer security[1] training and how this game contributes to improving the latter. The interested reader is encouraged to look at [Irvine1 2003] and [Johns 2004] for an in depth description of the game. The chapter depicts the contribution of this thesis to the overall goal of the CyberCIEGE project and provides a brief elaboration on those key concepts of computer security, the thesis focuses on.

A. CURRENT SITUATION IN COMPUTER SECURITY TRAINING

Within the last few years, the number of computer users increased continuously. Substantial progress in computer technology has made computers and network equipment available to a variety of users with different backgrounds and objectives. According to a survey by Jupitermedia corporation [Nua Internet Surveys 2004], the number of Internet users around the globe grew from 66.68 million, 1.61% of the population, in March 1998 to 580.78 million, 9.57% of the population, in May 2002. The number of Internet users in the US grew from 123.6 million, 45.33% of the US population in February 2000, to 165.75 million, 59.1% of the population in April 2002; a growth of around 7% per year. Computers are being used in various fields, ranging from simple text processing at home or at work, to e-commerce, Internet banking, and storage of customer or other sensitive information in databases accessed over networks, to complex data processing in technical and military environments.

With the increasing number of users, however, attacks on assets stored on computer devices or on the efficient use of IT equipment (i.e. by Denial of Service [DOS] attacks) have also increased. Hatcher [Hatcher 2001] reports that the combined deficit due to computer security breaches in 2000, amongst 249 US companies and agencies was about $265 million. In 2001, the deficit of 186 entities was nearly $378

[1] "Broadly speaking, security is keeping anyone from doing things you do not want them to do to, with, or from your computers or any peripherals."-William R. Cheswick, [March 31, 2004, from NCSA website: http://archive.ncsa.uiuc.edu/General/CC/ACES/workshop/tsld003.htm

million. This data indicates that more users rely on computers to store, or process assets of increasingly higher values. Therefore, computer security breaches cause increasingly greater deficits.

Due to frequent media coverage, the awareness of computer security threats, as well as countermeasures against attacks, have also tremendously increased. Many suppliers of web-based email are applying spam filters and virus scans automatically, and allow for user tuning. They are also providing virus alerts and supplemental information – i.e. SSL connections and child protection features [WEB.DE 2004]. Many other vendors provide low priced or even free anti-virus or firewall software. However, this increase has not translated into a significant change in user behavior. Users continue to use trivial passwords, fail to download security patches, or to use, and update anti-virus software. “Further still, corporate and government policy makers often elect to deploy weak protection mechanisms in environments subject to potentially highly motivated hostile attacks.” [Irvine1 2003]

Irvine [Irvine1 2003] and Johns [Johns 2004] deduce that the discrepancy between awareness and knowledge of computer security, and the failure to readily, continuously, and correctly apply computer security principles and measures in the real world, i.e. in every day life, is mainly affected by the type of computer security training available. The latter is described as often being “mundane and boring, for both users and administrators.” [Irvine1 2003] Having taken IT security classes and introduced new crewmembers to the ship’s security policy and measures as the responsible IT Security Officer, the author can confirm this perception.

The basic problem of any instruction or presentation is a human’s limitation in assimilating and memorizing a large amount of facts. Furthermore, many companies, but especially the military, have a high frequency of rotating people on a job. Thus, employees and soldiers are exposed to several differing and boring briefings. Each system is different; every department may have individual needs and possibly faces different threats. To meet the specific goals, policies and security measures are tailored to the needs of a ship, a department or an individual system. Thus, changing to a new

ship, department or system, requires new, potentially boring, computer security training.

In addition to “knowledge of the requisite facts”, some forms of engineering also require “a tacit understanding of the art of (…) engineering.” [Irvine1 2003] To make learning more interesting and more effective, i.e. to generate longer lasting knowledge, personal, “hands-on” experience may be necessary. “Sometimes people have to experience a problem in order to understand it.” [Irvine1 2003] In some areas military uses simulators to expose soldiers to above effects and to help them develop a tacit understanding. Both, military and commercial companies are recognizing tacit knowledge as a vital company asset and are searching for tools to measure and transfer it. [Richter 2003]

Despite an increase in computer security awareness, many users and policy makers still do not implement security principles in their daily life. Mundane and boring education and the lack of personal experience and tacit understanding might be a main cause.

What if education could be made more interesting, more hands–on?

B. A GAME TO TEACH COMPUTER SECURITY

A commercial-quality game for teaching computer security concepts could be used for both: to convey requisite facts and to generate tacit understanding of general concepts of computer security to a broad audience. [Irvine 2003]

Game play per se is interesting, as it stimulates a human’s natural behavior, his eagerness to explore. It therefore has the psychological advantage of guiding a student into a more open-minded state, compared to a potentially boring brief or classical classroom education.

Analogous to a simulator, a game creates a concealed virtual environment that allows the player to experiment and to test old and new ideas, without the risk of violating laws, or harming a real system, or in the case of computer security, without actually changing security settings and exposing the system to real threats.

A simulation or a game therefore has the potential to being both, time and cost efficient.

The CyberCIEGE game offers all these benefits.

> The game will simulate a range of scenarios involving computer networks. (…) The player will make security-relevant decisions about the network components and their interconnections. The player will also make decisions that affect the behavior of a set of virtual 'user' characters that perform other roles within the enterprise and must efficiently perform work for the player to succeed the game. [Irvine1 2003]

CyberCIEGE can be shipped with a set of starter scenarios [Johns 2004], but also offers the possibility for customers to write their own Scenario Definition File (SDF), tailoring the game to their specific needs. Thus, a teacher can emphasize special computer security topics and a company can train its employees for their particular IT environment. CyberCIEGE is able to serve a variety of customers: managers, administrators, teachers, students and regular computer users, in both, the military and the commercial world. It is extensible, to allow for the simulation of new threats and countermeasures, and it offers logging tools to aid instructors to review and analyze special situations, and evaluate student decisions.

The CyberCIEGE game offers an easily deployable and cost efficient tool, to teach about key concepts of computer security, while being interesting to play.

C. INTEGRITY IN A MILITARY LIKE FACILITY

Following statements taken from the webpage of different news agencies prove the increasing use of computer technology in military environments and also show security concerns and problems due to a lack of applying sound computer security principles. 'abcNEWS.com' posted following statement on their webpage:

> Tens of thousands of U.S. military and government computers containing sensitive information are easily accessible over the Internet, a computer security firm that cracked the networks said today. (…) Military encryption techniques, correspondence between generals, recruits' Social Security and credit-card numbers and other sensitive information is often stored on Internet-connected computers that use

> easily guessed passwords or in some cases no passwords at all, said an official at San Diego security firm ForensicTec Solutions Inc. [Sullivan 2002]

ISN posted following quote of WASHINGTON (AP) [3.22.99] on their webpage:

> The Command, Control, Communications, Computers and Intelligence systems, known as C4I, is compromised by security problems and also by a military culture prone to treating such problems as a lesser priority, the National Research Council reported. [WASHINGTON (AP) [1999]

Lt. Gen. Kenneth Minihan is quoted concerning attacks on military computers by 'wired news':

> Last year, over 250 unclassified DOD computer systems were known to have been penetrated. (…) The number of attacks are escalating; will double this year. [Minihan 1997]

Disregarding the use of simple office applications that are indispensable in any company, DOD and many branches of the military cumulatively use computers in sensitive areas, performing increasingly complex tasks. Today, most computers are at least connected to an Intranet. They are utilized for data acquisition, data storage, and data analysis and in many areas of the C4I structure. Marines use computers in land combat, they are contained in armored vehicles, serve as navigation and C4I equipment (i.e. TACOM). The Air Force has surveillance and communication networks, and today no modern aircraft would fly without the complex on-board computer systems. The Navy uses computers for similar purposes in ashore facilities, but also on ships, to enhance efficiency of their sensors and weapon systems.

The German Navy recently commissioned a new type of Frigate: type ship F124 "Sachsen". In terms of technology, this is one of the most modern battleships. The ship has a modular structure, modern sensors and effectors and it uses a digital network, the "databus-based integrated monitoring and control system (IMCS) – monitoring and controlling all technical systems and plants on board." [Blohm + Voss 2004]. (Figure 1 illustrates the complexity of the ICMS) The ICMS also provides several operating aids, fully automatic damage analysis and substantial simulation

capabilities. “All conceivable operating and damage scenarios can be simulated for crew training purposes.” [Blohm + Voss 2004]

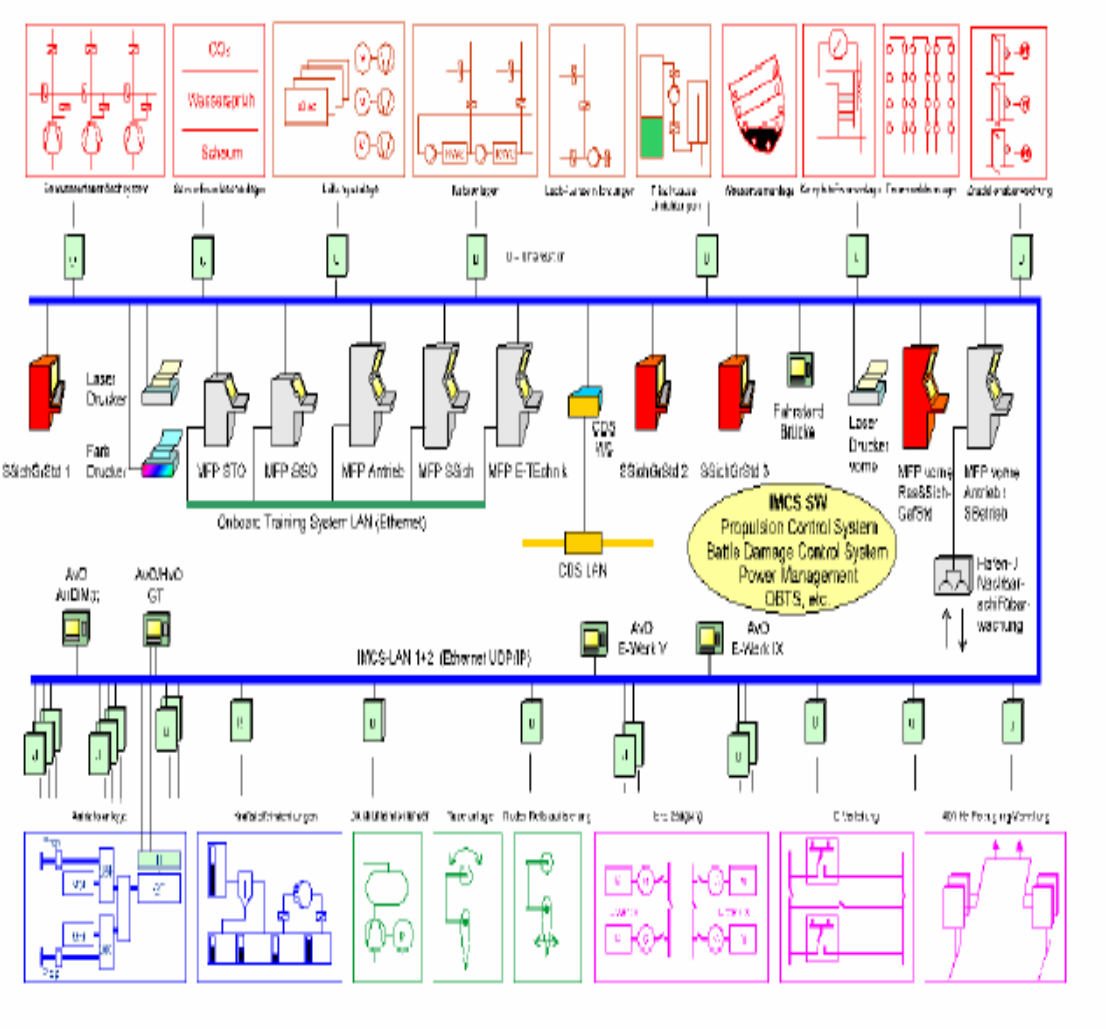

Figure 1. Integrated Monitoring and Control System [From Blohm + Voss 2004]

The most interesting innovation from a computer security standpoint, however, is “the decentralized combat direction system with the operational CDS [Combat Direction System] software, [where] computer capacity is distributed among 17 computers linked with a multiple-redundant ATM bus.” [Blohm + Voss 2004] Sensors and effectors are connected, analyzed and controlled by the system. The balance between operational ease and security measures in such and environment seems to be a

challenging task. (Figure 2 illustrates the complex networked environment of this combat system)

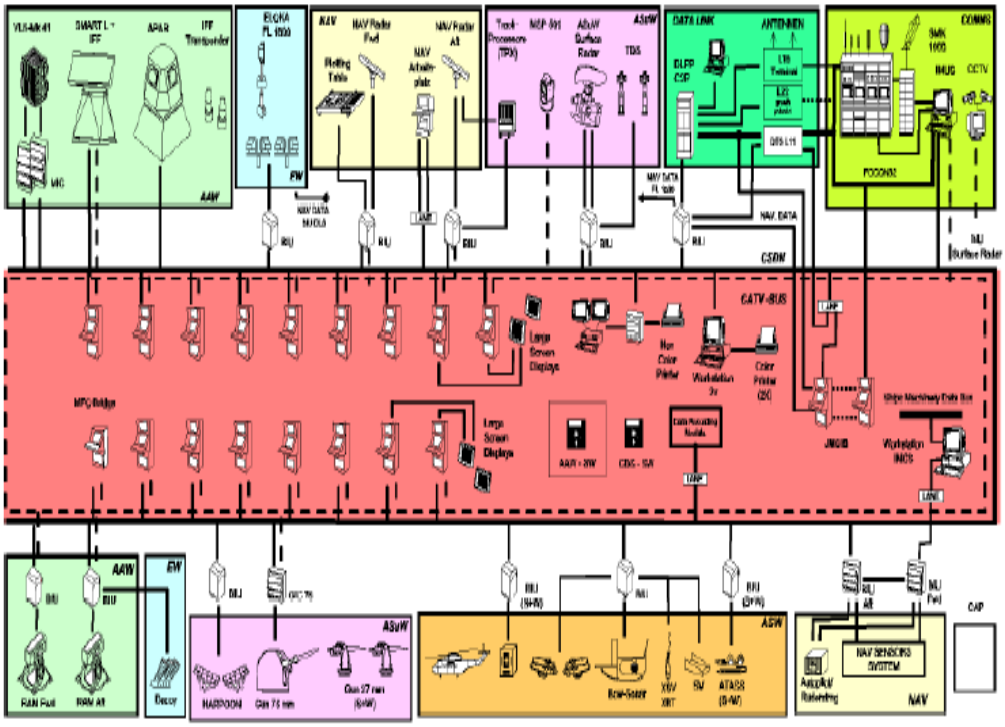

Figure 2. Combat System [From Blohm + Voss 2004]

It is evident, that ensuring computer security on such systems will be a major challenge for all groups, designers, managers and users, faced with the task to balance the needs for user convenience and operational needs on the one hand, and securing the systems, i.e. enforcing constraints imposed by computer security, on the other. What if an attacker succeeded in planting a trap door in the database software storing target information for missiles?

Motivated by the author's background as a Naval Officer, this paper concentrates on a scenario that creates a virtual world influenced by, and similar to a ship's Combat Information Center (CIC). The setting is a modern battleship with the need for high automation, thus sufficient computers and networks. The ship gains or loses money, depending on its ability to perform certain tasks. The player assumes the

role of the IT Officer, who has to provide components (workstations, servers, etc.) for his crew members, the users, and design a network to support the operational needs of the crew and their ship. The major challenge is a proper balance between user convenience and computer security measures that allows the player to implement and test his ideas about the best choice for the design of such a network, without actually setting it at risk. However, the scenario does not provide a detailed simulation of the shipboard environment. Instead, it is only a means of highlighting important computer security issues and raising their awareness.

Focusing on integrity of primarily data in storage, a computer security component that has become increasingly important in the military world [Ferraiolo 1992] [Clark 1987], this thesis contributes to the overall goal of the CyberCIEGE project, to teach users about different aspects of computer security, with a tool that is both, fun and educational at the same time.

D. KEY CONCEPTS

For readers with no or only minor computer security background, to better understand the topics addressed in this thesis and the implications of educational goals and test cases, this section describes some of the underlying key concepts used in this paper. The set presented here is, however, not intended to be complete for either the CyberCIEGE game engine, nor for the scenario developed by this thesis.

1. Computer Security

The notion of computer security is the basic concept the IT Officer on a highly computerized ship, such as that described above, has to understand, in order to identify the type, and importance of the ships assets, to correctly assess the threat posed to those assets, and to analyze the security measures, to ensure their protection.

Computer security, however, is a broad term, composed of other key concepts. The National Information Assurance Glossary [CNSS No. 4009] defines computer security as:

> Measures and controls that ensure confidentiality, integrity, and availability of IS [Information Systems] assets including hardware,

software, firmware, and information being processed, stored, and communicated.

According to Brinkley and Schell "there are many characterizations of computer security." [Brinkley 1995] The definition used in this paper relates to information technology security as defined by the European Community, and cited in [Brinkley 1995]. Computer security is composed of three properties:

Confidentiality: Prevention of unauthorized disclosure of information.

Integrity: Prevention of unauthorized modification of information.

Availability: Prevention of unauthorized withholding of information or resources.

The distinction between computer security and information assurance is, that the latter incorporates the notion of authenticity[2] and non-repudiation[3] in addition to the properties mentioned above. [CNSS No. 4009]

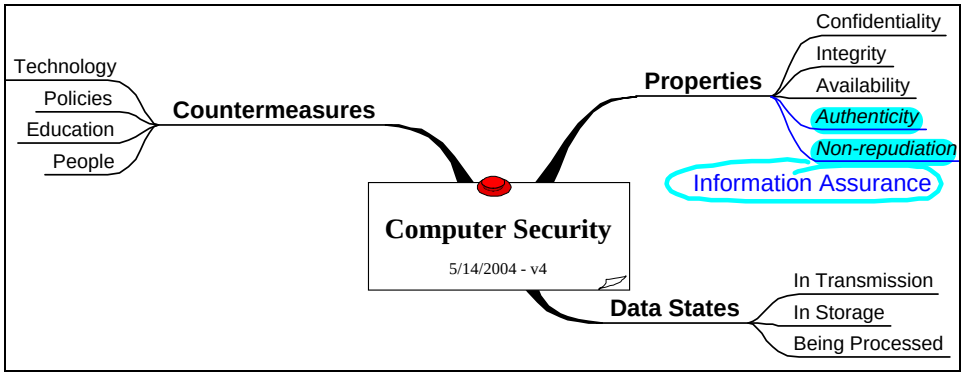

Figure 3. Computer Security Versus Information Assurance [After Usher 2003]

The CyberCIEGE game addresses all properties of computer security. Several scenarios are being developed with various levels of complexity and with different objectives. Some of the scenarios focus on only some properties, e.g. only on physical security, or mainly on confidentiality. The combination of diverse scenarios

[2] Authenticity is the notion of assuring that the sender of information, i.e. a message, is indeed its originator.

[3] The sender and recipient are provided with proof of the transaction, so none can deny to be part of it. [CNSS No. 4009]

contributes to the overall goal of the game. This thesis concentrates on the aspects of integrity to a military-like facility.

2. Confidentiality

When speaking of security of military data, most people automatically think of the challenge to prevent sensitive data from being disclosed to unauthorized personnel, be it the adversary or the media. Including the notion of electronic data processing, 'unauthorized personnel' is expanded to 'unauthorized processes, or devices'. [CNSS No. 4009] [Clark 1987]

A battleship has to handle several types of sensitive information on various classification levels. Although most messages are at lower levels of classification, there are, however, several confidential, and some secret or even top-secret ones. Information concerning force disposition, weapon or sensor parameters and profiles, and communication codes, all need to be thoroughly protected against disclosure. The scenario generated by this thesis addresses some aspects of confidentiality and provides assets of different security clearance levels. However, most of the direct users within the simulated world are cleared at a level equal to that for authorized reading of all information[4]. Scenarios developed by other students address confidentiality more closely.

3. Integrity

Although Clark [Clark 1987] and Ferraiolo [Ferraiolo 1992] state, that the military is mainly concerned with protecting the confidentiality of information, and that it is mostly the commercial world which has to deal with integrity, they conclude, "Indeed, much data processing within the military exactly matches commercial practices." However, they still note a "difference in priority."

Considering the fact, that the military increasingly deploys systems depending on computers mostly connected to networks, a significant increase in the military's interest in the integrity aspect of computer security can be noted. What if the software controlling the reactors of nuclear submarines or aircraft carriers contained malicious code or flaws? What if the targeting data was modified while entered into a missile's

[4] There are some indirect users, e.g. the Internet, who lack adequate clearance.

navigation system? What if an attacker could modify the entries in the weapons or electronic warfare database of a battleship?

Reportedly, US Intelligence was able to insert a Trojan Horse into the software, acquired by the K.G.B. to control critical components of a gas pipeline. The Trojan Horse was preset to cause a system malfunction. “The result was the most monumental non-nuclear explosion and fire ever seen from space.” The explosion occurred in June 1982. [Safire 2004] The resulting loss of the Soviet Union’s trust in western software is since known as ‘The American Software Problem’. [Irvine4 2004]

Ken Thompson described a trap door in an early version of an UNIX compiler, introducing a copy of itself into any software that it compiled. [Irvine 2 2002]

There are many questions regarding the assurance of systems:

How much commercial off-the-shelf software is used in military applications? Might a terrorist group be able to insert a trap door in software that is a part of a critical military system? What about possible mistakes a cleared user makes, while handling sensitive data? What about regular flaws experienced by nearly any computer user on a regular basis?

This thesis focuses on the integrity aspect of computer security. The National Information Assurance Glossary [CNSS No. 4009] defines integrity as:

> The quality of an IS [Information System] reflecting the logical correctness and reliability of the operating system; the logical completeness of the hardware and software implementing the protection mechanisms; and the consistency of the data structures and occurrence of the stored data. Note that, in a formal security mode, integrity is interpreted more narrowly to mean protection against unauthorized modification or destruction of information.

Sandu and Jajodia [Sandu 1995] suggest using the term “improper” instead of “unauthorized”, since modification can happen even without authorization violation. An authorized user can modify information as a mistake or deliberately, if he responds to social engineering.

Irvine [Irvine2 2002] distinguishes between integrity of data in storage and that in transmission. Integrity of data in storage can further be subdivided into integrity of

data, and integrity of source code. The latter has different implications, whether it is used on a component or a system. Being on a system, its integrity level defines the maximum integrity capacity of the system.

A modern battleship is equipped with several systems that are exchanging sensitive information with other ships of the force and the head quarters, e.g. information about the weapon status or force disposition. Integrity of the data while in transmission is mostly preserved using secure lines, i.e. encryption devices. However, if the ship uses an internal network, data integrity might be prone to attacks, if the network spans zones of different protection and classification levels. A malicious crewmember might be able to wiretap the less protected line.

Considering that a battleship needs to handle several databases containing sensitive information, e.g., a database with weapon parameters or electronic signatures of hostile missiles, or a database containing personal and medical information of crewmembers, integrity implications of data in storage are of high importance. IT personnel need to carefully consider where to store the data, how to secure it, and who to grant access to the components holding the asset. To make the data available to subsystems and users, careful analysis of the environment, attached components or networks, is required to balance the need for fast and easy access, yet sufficient protection of the data integrity. For example, how would the reaction window of the ship in applying countermeasures against an incoming missile, flying at supersonic speed, be affected, if the electronic signatures were stored on a stand-alone component? On the other hand, consider the vulnerabilities that result by connecting the database to the network, which is used to grant access to off-the-shelf office applications, or even worse, the same network that allows navigational personnel to access the Internet to retrieve current weather updates. This latter choice of architecture would expose the data to low integrity software, potentially containing trap doors or even to hacking attempts over the Internet.

The ability to protect the integrity of sensitive data in such a scenario depends on the right choice of the integrity of software used to store and operate on the data,

and a proper evaluation of the integrity capacity of the component to store the high integrity assets, and the network used to facilitate access these assets.

4. Availability

An important aspect of military security measures is to ensure that all vital systems are fully operational when they are needed. Some need to be running 24 hours a day, seven days a week, while at sea. Ensuring uninterrupted use of its systems, i.e. protecting against Denial Of Service (DOS) attacks, is vital for most military systems. On ships, navigation and propulsion systems are vital for safe maneuvering. An engine breakdown or the inability to correctly determine its position while e.g. passing tight waters or a cleared mine channel might result in a collision or explosion, endangering the lives of hundreds of sailors. Were attackers able to initiate a system crash of the Combat Direction System (CDS) of, for example, the F124, during an engagement, while under aircraft or missile attack, the ship would be defenseless.

5. Assurance and Software Integrity

In the context of computer security, assurance is the level of confidence that the security features, architecture and procedures of a computer system correctly enforce the security policy. [CNSS No. 4009] It is the trust in the software engineers, programmers, and the development process, to ensure that the software fully meets its requirements, and has no flaws, no trap doors and no additional functionality, and no harmless or malicious Easter eggs. The ability to demonstrate these properties is provided by evaluation assurance levels (EAL) from the Common Criteria (CC) nomenclature.[CC 2004] The CC provides detailed guidelines on tools and techniques to be applied by developers and evaluators. To demonstrate that, for example, a security kernel[5] (SK) is free of malicious code, the tools and techniques requested for an assurance level of EAL 7 have to be applied.

However, these tools and techniques do not apply to other arbitrary software, including applications, and general-purpose operating systems (OS). This is mainly, because most applications and operating systems are much too complex to be analyzable. Therefore, these applications and OSes cannot be evaluated at high assurance levels, i.e., not at EAL 6 and EAL 7.

[5] See paragraph II.C.6 for a definition.

The CyberCIEGE game engine uses a value to represent the strength of applications and operating systems. OSes that have been evaluated to enforce a policy with high assurance are given high values. Additionally, applications and OS that have been developed in a closed environment, receive a high value.

To address the strength of software that does not enforce a security policy, this paper uses the notion of software integrity. High integrity software has a high strength value. With an increase in value, the likelihood that software contains a trap door or a Trojan Horse decreases.

When assessing the likelihood of attacks on the integrity of an asset, the game engine picks the lowest strength value of the OS and all the applications of the component that have access to the asset.

In the case of operating systems with a MAC enforcement mechanism of high assurance, the OS has a high strength value. This high value has two effects:

1. It indicates that the OS is less likely to contain malicious code, which directly affects the integrity of the data. The likelihood of data modification decreases with an increasing strength value.

2. The OS is able to enforce a policy to protect the integrity of the data, even in the presence of low integrity data or low integrity software on the same component.

The integrity and the assurance of software, applications and OSes, is reflected in the description provided inside the encyclopedia of the CyberCIEGE game. The assurance is indicated using an EAL from the CC nomenclature. The integrity of software is indicated by descriptive text. Very high integrity software is described as having been produced in a closed environment. Software of low and moderate integrity is described in terms of thoroughness and amount of testing, and in some cases, by evaluations at low and moderate EAL.

The tuning of the game engine offers the opportunity to distinguish between very high integrity software, and other software. The tuning, however, does not provide the means to distinguish between moderate and low integrity software.

The level of confidence in the software and the OS used on a system clearly will influence its integrity capacity. [Irvine2 2002] Thus, if the IT Officer of a battleship chooses to run both high and low integrity software on a system that has no security kernel the maximum integrity capacity of the system will be bounded by the capacity of the component with the lowest confidence: the low integrity software, if the OS is of higher integrity, or the OS, if it is of lower integrity. This statement holds, regardless of which software is dedicated to process the data. It is expected that there is a high probability for low integrity software to contain malicious code. Therefore, if the IT Officer chooses the low integrity software to process the data, the malicious code inside the software will have direct access to the data and will modify it. If the high integrity software is chosen to process the data, the malicious code inside the low integrity software will still be able to access and modify the data in the absence of a SK, even if it is not dedicated to operate on the data. If the OS is the lowest integrity component, malicious code within the OS will also have direct access to the data. Therefore the IT Officer might not be able to preserve the integrity of high integrity assets on such a system.

6. Reference Monitor / Security Kernel

The development of the Reference Monitor (RM) concept was, according to [Schell 1995], primarily motivated to counter the threat posed by the potential presence of malicious software.

> A Reference Monitor 'enforces the authorized access relationships between subjects and objects of a system.' [Irvine3 2004]

The RM needs to be:

a. Tamperproof,

b. Always invoked,

c. Analyzable.

An implementation of the RM concept is a security kernel (SK).

Using a SK would allow an IT Officer to compose a multilevel system, i.e. a system allowing access to personnel of different clearance levels and allowing storage and processing of data of different classification levels. Such a system would also

allow the IT Officer to install applications of different integrity levels on the same system. Properly configured, the SK would grant the high integrity application, which is dedicated to modification of the high integrity asset, access to the asset, while denying access by other applications, i.e. low integrity software dedicated to other purposes or possibly a Trojan Horse within that software.

7. Closed Environment

In the *Glossary of Computer Security Terms*, the National Computer Security Center defines the term 'closed environment' as follows:

> An environment in which both of the following conditions hold true:
>
> (1) Application developers (…) have sufficient clearances and authorizations to provide an acceptable presumption that they have not introduced malicious logic.
>
> (2) Configuration control provides sufficient assurance that applications and the equipment are protected against the introduction of malicious logic prior to and during the operation of system applications. [NCSC 1988]

Several military systems are produced in such a controlled environment, employing trusted programmers. In the game, the player needs to buy software, and choose between different applications. Some of them are produced in such a closed environment.

8. Intranet – Internet

A network that connects components or networks of an organization, while disallowing access to components and users that do not belong to this organization is considered an Intranet. The network on board a battleship, connecting the components of the ship would be the ship's Intranet. The military network, allowing access only to military components would be the Military Intranet. Examples are the Navy Marine Corps Intranet (NMCI) and the SIPRNET.

For the purpose of the CyberCIEGE game, the network known as the World Wide Web, allowing access to anybody is considered the Internet. In the game, there are no means to keep attackers from accessing the Internet. However, the player can make choices to prevent attackers from accessing an Intranet.

9. Trojan Horse

Hidden malicious code within software that offers a useful service to its user is referred to as a Trojan Horse. This malicious code performs additional tasks, hidden from and not intended by the user, i.e. “allowing the unauthorized collection, falsification, or destruction of information.” [CNSS No. 4009] Subversion is hidden code intended to undermine, or circumvent the protection mechanism of a system. A Trojan Horse differs from subversion in that it requires the cooperation of the victim, i.e. the victim has to run the application containing the Trojan Horse. Thus, the attacker cannot entirely choose the time of its activation. Because the application runs on behalf of the victim, the Trojan Horse is constrained by security controls imposed on the victim. Malicious code falling into the category of subversion is independent of a victim’s cooperation. It bypasses the security controls and is activated and deactivated by a trigger mechanism, allowing the attacker to determine the time of execution. While a Trojan Horse “executes in an application”, subversion “may execute within the OS,” [Irvine3 2004] An example of a Trojan Horse may be a small, popular game that a sailor, having access to the Internet downloads freely from the website of an attacker. While he is playing the game, malicious code within the game code is scanning the files on the system for sensitive information, e.g. communication codes, or is accessing and modifying the Electronic Warfare Signature Data Base (EWSDB), provided the user is connected to the same network, the CDS, and has the required clearance to access the assets. What if the ship’s CDS was connected to the Intranet and had no SK, i.e. no Mandatory Access Control (MAC)? Then every sailor able to introduce software to the Intranet could endanger the ship’s safety.

An example for subversion would be malicious code within the OS of the Electronic Warfare System, which processes radar signals and has modify access to the EWSDB. An attacker could feed the system with a deliberately constructed radar signal that triggers the activation of the malicious code, which would then result in modification of crucial data. A trap door can also be seen as an instance of subversion.

The CyberCIEGE game models both, subversion, and the notion of a Trojan Horse. Within the CyberCIEGE game, the player is provided the choice between

different types of component / OS combinations. Subversion is modeled, by providing operating systems with low integrity MAC enforcement mechanisms. These OSes have a high likelihood of attacks triggered by the game engine. Trojan Horses are contained in low integrity applications. Within the CyberCIEGE game, the player is provided the choice of installing a particular application on a component, by purchasing or selling software. The choice of the application's integrity level directly influences the likelihood of a Trojan Horse attack. However, the game does not provide the granularity to model the player's choice of running a particular application, to influence the activation of a Trojan Horse.

10. Trap doors and Subversion

The National Information Assurance Glossary [CNSS No. 4009] defines a trap door as a "Hidden software or hardware mechanism used to circumvent security controls." The expression is synonymous with a back door. A penetrator seeking a means to gain access to a target system that is virtually undetectable and therefore not subject to patching, will try to install a trap door into the system. [Karger 1974]

The difference between a trap door and subversion is, that a trap door is an instance of subversion. The trap door considers hidden code that allows an attacker to circumvent security controls, to gain access to a component. Subversion is the more abstract term, incorporating additional attack types, for example those attacks, where the malicious code is triggered to modify crucial data, without the need for an attacker to get access to the component.

Three major classes of subversion can be distinguished, depending on the point of insertion into the software:

a. Insertion into the software can be performed at the production facility.

b. Insertion can be performed during software distribution, either as part of the initial distribution, or as updates. An attacker can intercept and modify the software during transport, or he can generate bogus copies containing malicious code and distribute them to customers.

c. Insertion can be performed during installation and operation of a system. [Karger 1974]

All of these types of subversion are a threat to military systems. Due to budget constraints, the IT Officer of a battleship might need to buy low integrity software, thus potentially introducing subversion inserted at the production facility. Today, no major system is really free of flaws. Most military system will need frequent updates to meet changes in user and operational requirements. Both, regular and embedded systems frequently need technical assistance by company personnel, to perform initialization, updates or maintenance of the system or its components. The notion of subversion will be a major challenge in the scenario developed by this thesis.

E. SUMMARY

With the increasing number of computers used and interconnected in DOD, businesses and private homes, attacks on valuable, electronically stored assets have grown, too. Despite heightened security awareness, computer users and policy makers still tend to choose weak and insufficient security measures. This discrepancy is mainly affected by inefficient, primarily because it is boring, security training. CyberCIEGE provides hands-on experience and a tacit understanding about security implications, while being fun to play and consequently can overcome the problem of today's security education. This thesis contributes to the game's overall goal by creating a scenario that introduces the player into a virtual military-like facility and teaches security concepts focused on the integrity of data in storage.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SCENARIO GOALS

This chapter discusses the group of intended players and the educational goals pursued by the thesis.

A. INTENDED PLAYERS

Trying to change the discrepancy between security awareness and failure to apply security principles in the daily life, there are four approaches differing in their time of impact. Educating personnel working as IT professionals, e.g. administrators, yields the fastest results. Training the future workforce certainly has a greater latency. Familiarizing decision makers and managers ranges in between the above margins. The first addressee is DOD personnel. However, cooperating with a commercial company and planning to bring the game to the open market, commercial and home users are also considered to be potential players. One should keep in mind, that DOD recruits many of its employees from universities and the civilian labor market. Thus, today's commercial and home users might become future DOD personnel. Investing in the computer security education of this group will not only improve the global computer security situation, but will provide DOD with employees who already have a good basic understanding of computer security. This will reduce DOD's employee training costs.

1. IT Personnel

IT personnel form the group that has the fastest means to apply security principals. They literally have their 'hands on' the devices that need to be implemented and configured in a way to properly enforce security principles. They are responsible for updating and patching critical systems, for configuring routers to block ports commonly known to be used by malicious software, for disallowing ftp, and for monitoring and performing audits on important network traffic. IT personnel have the means to set up their networks in a way to form secure cells that guard users from outside attacks, and also contain internal users or computers to launch attacks on others, using the Internet or connected networks. In other words, they can respond the fastest to the security principles learned by playing CyberCIEGE.

IT personnel can use CyberCIEGE as an educational tool to train new employees in either the IT department, or, if it complies with the company policy, any employee who is to be given access to company computers. Thus, they can generate a pool of employees who understand the basic security principles and therefore show more understanding for imposed security constraints; e.g. for the denial of installing privately owned software or the imposed password length and the need for frequent password changes. Employees trained by the game, will most likely have a better tacit understanding of security concepts and consequently reduce their error rate.

IT personnel can use the game to create a virtual model of their network topology, serving several purposes:

a. To train employees in a more focused fashion.

b. To perform limited tests of basic principles of their current design.

c. To plan and test changes before implementation.

d. To present their plans and designs to management and illustrate cost benefit considerations[6].

2. Management

The benefit of educating decision makers about computer security is that they realize the need for issuing a security policy and how their policy affects the company. The latency for results to show effects might be longer compared to the IT personnel, but once in place, they have a much higher potential to encourage or even enforce changes.

Playing CyberCIEGE, managers can learn about the basic computer security principals, such that they can apply them in their interaction with devices. However, much more important is the opportunity to develop a tacit feeling for the implications security policies and imposed or neglected security measures have on a company's assets and might have on its employees. Security measures come with a price, but the financial impact of neglecting them might far outweigh the investment.

6 Note, the game is not a simulator. Only basic concepts can be illustrated and tested.

Consequently, knowing about security implications, managers and decision makers can develop better security plans, can make better decisions about funding of security measures, and will most likely be more successful in mobilizing their employees to follow their guidelines.

3. Students

Students form the long-term investment group for CyberCIEGE. Students eventually graduate and enter the labor market and will work for the DOD and companies most likely using IT systems as programmers, system developers, software engineers, administrators, evaluators, System Certifiers or Security Officers. Sooner or later, some might reach the management level. However, investing in a sound computer security education for this group will have the greatest impact in the long-term perspective.

The primary aim of the game is to provide educational facilities with a tool that can be used in laboratories to supplement computer security class education. [Thompson 2004] In laboratories students can deepen their understanding of the principles learned in class, by playing simple scenarios that synergize or build on each other. They can experiment with different policies and with different ways to implement such policies and are given a virtual world to develop and test, in a limited way, new ideas easily and with cost efficiency. Most important, however, is the opportunity to develop a tacit understanding of the underlying basics of computer security, about the benefits and drawbacks of policies, and about the advantages of different strategies to solve problems that have no single correct solution. These problems include the necessity to mediate between the ease of operational use of computer systems versus security confinements and benefits of security measures versus the cost to implement them. While devices and prices change with high frequency, the basic principles are steady. This type of knowledge will last much longer than any device-specific education.

4. Instructors

Instructors can use CyberCIEGE to assign work in a laboratory, as a supplement to their lectures. They can use the features of the game to evaluate the

progress of students and to analyze whether there are ideas and concepts that are difficult to grasp and need in-depth elaboration in class.

Teachers might use the scenarios that come with the game or develop scenarios of their own, to illustrate focal points and emphasize special security implications.

Learning basic computer security principles is vital for both the corporate world and DOD. In both, personnel responsible for managing computer technology and those using and working with computers need to be familiar with these principles. Although managers, administrators and normal users apply knowledge about computer security in different ways, they all can use the CyberCIEGE game to learn basic computer security principles. The game can be especially beneficial in the educational environment, to help instructors and students to achieve their goals and thus providing DOD and companies with employees skilled in computer security matters.

B. EDUCATIONAL GOALS

The previous section identified various groups as intended players. The groups apply knowledge of computer security principles in different ways. Managers use their knowledge to develop sound security policies; administrators and IT personnel, to implement the policy and build sound architectures; and employees use their understanding to apply security principles in their daily use of computers. The underlying basic principles of computer security, however, are common to all groups. The goal of the CyberCIEGE project is to teach all aspects of computer security. This paper focuses on integrity, as a contribution to the overall goal. This section analyzes the major risks to data integrity and outlines the four major educational concepts to be taught to the player.

1. Trap door – The Low Integrity Software Problem

The first and most obvious risk, in terms of data integrity, can be referred to as the trap door problem caused by low integrity software. (Figure 4) In this case, a user who needs to modify data of high integrity chooses to use software of low integrity. Low integrity software corresponds to off-the-shelf applications or operating systems, i.e., those types of software bought from a vendor and programmed by programmers

neither connected to, nor personally known by the user. In the case of military systems, these programmers are civilian personnel often working without any background checks or military clearance. Usually, no information is available to the prospective user about the principles applied during the software development process, nor is sch software reviewed or evaluated by a third party, for example in accordance with the Common Criteria, before installation. Thus, there is no reason to assume, and certainly no guarantee, that the software is flawless or free of malicious code.

The resulting problem for the user is the risk that this software might indeed contain malicious code, such as a trap door, as an instance of a subversion mechanism. To trigger the trap door, an attacker can use any data sequence that is part of the admitted input domain of the system. This sequence serves as the signal recognized by the trap door to activate or deactivate the subversion mechanism, and e.g. grants the attacker root or operating system access to the system, allowing him to modify the high integrity data. If the subversion mechanism resides inside the module that verifies the input for validity, even a specially crafted data sequence that is not part of the admitted input set can serve as a trigger. This is because every data sequence submitted to the system needs to be read and evaluated prior to the decision to accept or reject it. Thus, the subversion mechanism can read and recognize the activation / deactivation signal.

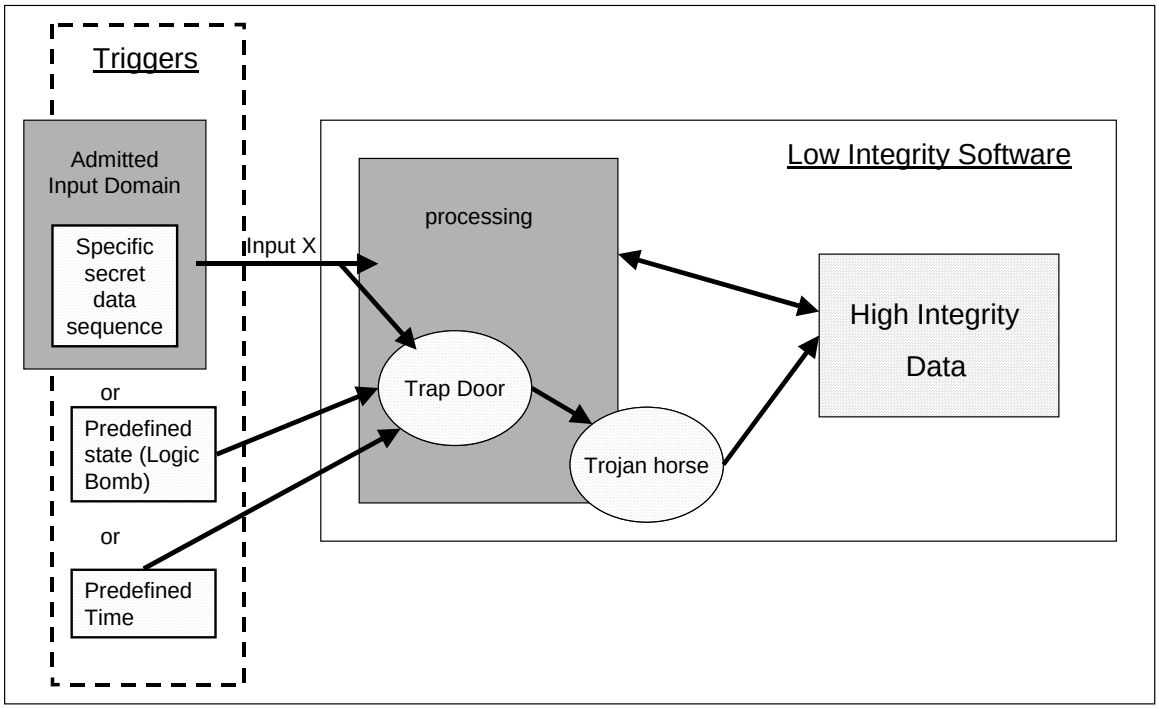

Figure 4. Low Integrity Software Problem

In the case of a Trojan Horse, the malicious code runs every time the harboring application is executed. Running on behalf of the user, the Trojan Horse has the same privileges as the user and can therefore access and modify the high integrity data.

This modification of data can damage financial or critical military or medical environments, if the user does not recognize it. Recognizing an unauthorized modification, however, is a difficult task. The trigger might be a certain combination of keystrokes, or a specific pattern of radar signals used in analyzing systems of modern war ships. In the case of low integrity software, since users do not analyze the source code of a program, a valid trigger might as well be a time bomb, in which the malicious code starts to execute at a pre-defined date.

The major educational objective concerning this risk is for the player to realize that in order to protect high integrity assets he has to invest into high integrity software. Using low integrity software for either the OS, or the application accessing high integrity assets, poses a high risk to the asset's integrity.

2. Trap door – The High Integrity Software Problem

The next risk to consider is the problem of evaluating software at a high level of assurance. To avoid the trap door and Trojan Horse problems discussed above, a user who needs to modify high integrity data decides to use high integrity software. (Figure 5) The user now faces the problem of determining the level of assurance with respect to the software's ability to preserve the integrity of the data. Today, the solution to achieving assurance for off-the-shelf software is to have it evaluated by a third party. The process of evaluating even relatively simple applications, however, is very time consuming and expensive. Furthermore, today's applications consist of very many 'lines of code' and are mostly of a structure that does not separate 'security relevant' from 'other code'. Consider, for example, the operating systems of Microsoft, or applications like Word or Power Point. Most applications lack any specification of what 'correct behavior' actually is. If a detailed description of the required functionality is missing, the notion of having 'no additional functionality', and thus no malicious code, is meaningless. For such software an evaluation at EAL7 is actually not possible. So there can be no guarantee that the application does not contain a well-programmed and hidden Trojan Horse.

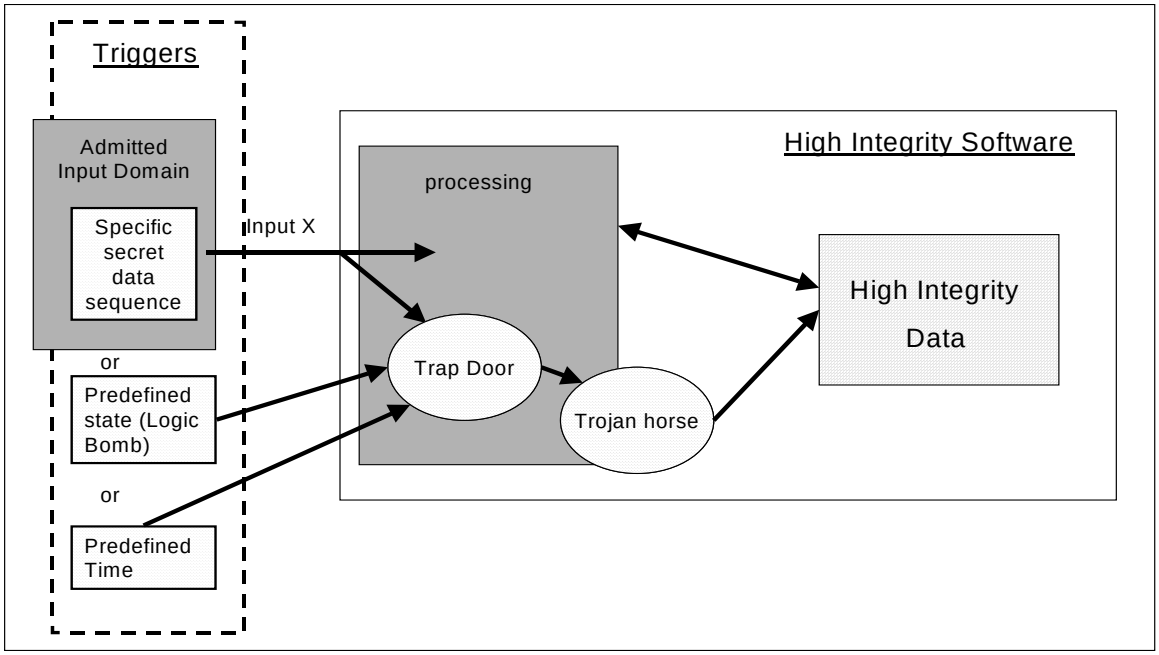

Figure 5. High Integrity Software Problem

One problem is that a skilled attacker might be able to hide a Trojan Horse consisting of only a small amount of code such that it is not found even in the process of a thorough third party high assurance evaluation. This could be achieved, for example, by programming 'dual use' code as an integral part of application performance which, given an expected input, performs accurately, without any sign of a possible irregular behavior. Given, however a trigger that is a predefined input sequence which is a particular, possibly very rare subset of the admitted input domain, this code then maliciously modifies the high integrity data. As an example, consider the domain of admitted inputs to be any radar signal intercepted by a sensor and forwarded to an analyzing system. Let the trigger be a particular radar signal, of all radar signals possible. The extent of this input domain is well beyond the limits of exhaustive[7] testing for current systems. Thus, by providing special data, the trigger, an attacker is able to control the behavior of even a thoroughly tested and evaluated application and alter the data that ought to be protected. For example, Anderson [Anderson 2002] demonstrated how a specially crafted corrupted UDP packet could be used to activate and deactivate a small artifice within an operating system supporting standard Network File Server code.

An additional problem related to software evaluation is the "Turing machine" behavior of even moderately complex applications. This illustrates the problems that exist even in the case of closed environment software, which is produced by trusted programmers. In this case, it can be expected that the software is free of Trojan Horses. Again, even a third party evaluation cannot absolutely guarantee that the software will preserve the integrity of critical data. If the attacker is able to provide input data that is not a subset of the admitted input domain, the application might transit into an unpredictable state, essentially performing instructions scripted by the attacker. Thus, it would no longer be able to preserve integrity.

[7] Testing of every possible case

The major educational objective concerning this risk is for the player to realize that

a. Security has a price: It is costly to develop software to high security standards.

b. Claims by vendors about security features and high integrity are not trustworthy, if a trusted party has not evaluated the software.

c. Some software can be evaluated at high assurance levels. This is software that is simple enough to be analyzable. This software can be part of a high integrity system. Some applications are too complex, to be analyzable. Thus, it is not possible, to achieve high assurance evaluations on this software. In this case, a means to achieve high integrity software is to produce the software in a controlled environment, with trusted programmers. (Although some complex systems can be restructured to separate security-critical from non-security-critical functions, this does not help, if the non-security-critical functions must operate on the high integrity data)

3. MAC Enforcement Mechanism

A further problem, regarding integrity of stored data, is called the MAC enforcement mechanism attack. (Figure 6) This case illustrates the limitations of a MAC enforcement mechanism's ability to protect high integrity data in a multilevel system. A system designed to handle multiple levels of security, i.e. both, high and low integrity data, uses a MAC enforcement mechanism to mediate the subjects' request to access objects. The task of the MAC enforcement mechanism is to guarantee that only subjects with the required security label are permitted access to an object. Suppose the multilevel system incorporates two applications of different integrity. Application A1 is designated to operate on the high integrity data and is a high integrity product, free of malicious code, and most likely very expensive. Application A2 is assigned to operate on less critical, low integrity data and is off-the-shelf software of unknown origin, most likely much less expensive. A1 is allowed modify access to the high integrity data, A2 is allowed read access only.

If the MAC enforcement mechanism itself is not a product, designed to be evaluated at high levels of assurance, it will most likely have flaws. An attacker, who

placed malicious code into A2, can exploit the flaws of the MAC enforcement mechanism and allow A2 write access to the high integrity data. Similarly, if an attacker was able to place malicious code inside the MAC enforcement mechanism, he would also be able to alter the high integrity data.

Consider when the MAC enforcement mechanism is a security kernel (SK). It is trusted to be free of malicious code. In this case, it can reasonably be expected that the access control mechanism protecting the data cannot be circumvented by an application. However, there is still a risk to be addressed: the risk of improper implementation. If the low integrity software, A2 is assigned to operate on the high integrity data, A2 will be assigned the required permission, i.e., modify, and thus, the SK will grant A2 modify access to the data. In this case, an artifice inside A2 is able to modify the high integrity data.

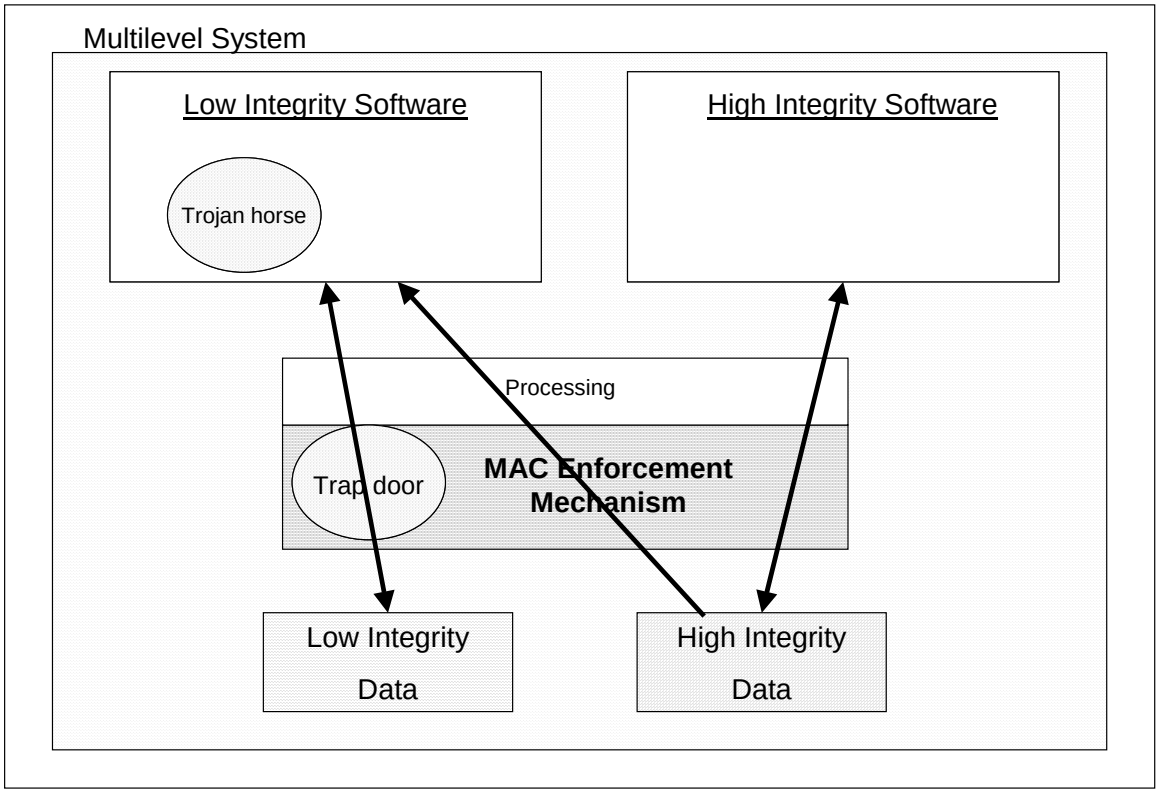

Figure 6. Attack On MAC Enforcement Mechanism

The major educational objective associated with this risk is for the player to realize that

a. It is important to choose a trusted MAC enforcement mechanism, thus a security kernel, to preserve the integrity of high integrity data in multilevel systems, which use software of various integrity levels.

b. It is important to properly implement the RM concept. A SK will not be able to protect high integrity data if the security implementation assigns low integrity software to access the data.

4. Social Engineering Attacks

If a system is constructed, evaluated and implemented thoroughly, and considered secure with respect to technical computer security, there is still one more risk to be considered: the so called social engineering attack. (Figure 7) This attack addresses the issue of trust in operational personnel.

In this case, the high integrity data is stored on a dedicated system, with no connectivity to external data, or networks. The application used to store and modify the data is a closed environment product of high assurance, posing no threat to data integrity. The only remaining variable available to an attacker is the personnel trusted to operate the critical data. The protection of the data depends solely on the integrity of the trusted user.

The means an organization or the military uses to assess the level of trustworthiness of an employee is a thorough background check. The degree and accurateness of background checks is a question of financial effort, the timeframe for its performance, and the intended clearance level. An initial background check, however, will not suffice to guarantee high trustworthiness over an unlimited time frame. Periodic reassessments of those employees trusted with high security and integrity data are necessary to reduce the threat of subversion.

The value the protected asset has for an attacker is of great influence. If it is high enough, he will most certainly devote sufficient energy and finances to find overwhelmingly tempting incentives customized to a trusted employee with access to this asset. The employee will then carry out the malicious action. In addition to background checks and periodic updates, to ensure the security of its assets, a

company needs to have some means of auditing[8] and of punishment that pose at least as high a threat to a trusted employee, as the incentive offered by the attacker.

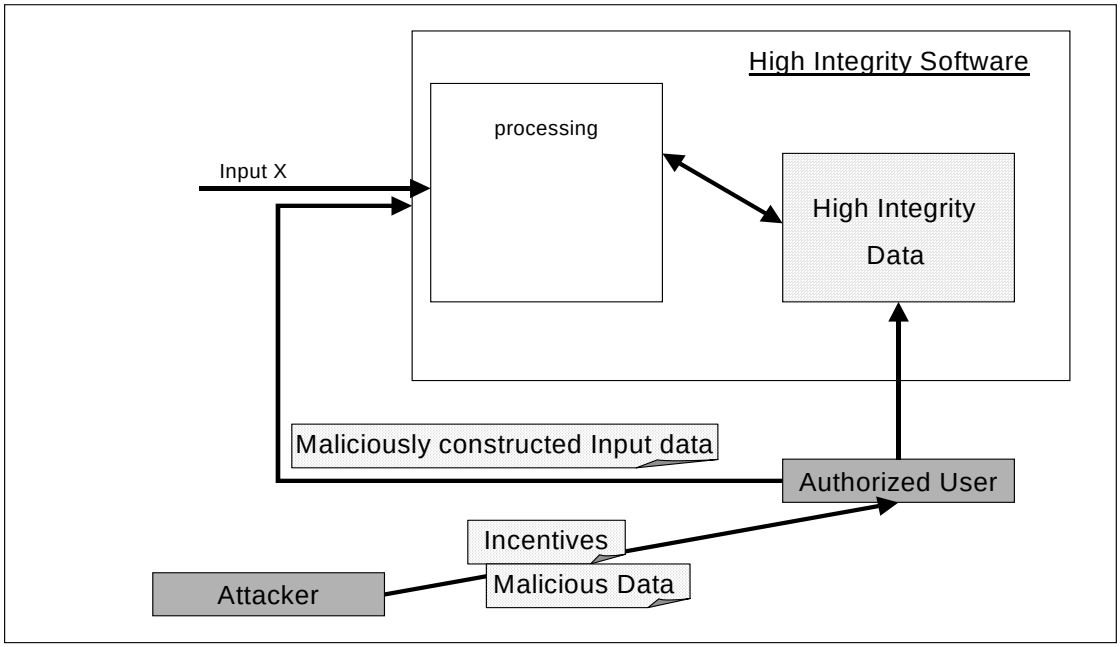

Figure 7. Social Engineering

The major educational objective concerning this risk is for the player to realize that security measures include more than technical measures. It is vital, to include the notion of trustworthiness of personnel in security considerations for the system.

C. SUMMARY

The CyberCIEGE game primarily addresses current and future DOD personnel as the intended players. The scenario definition file (SDF) created by this thesis aims at educating four major groups: IT personnel, management, students, and instructors. The thesis concentrates on teaching four major educational goals, while focusing on integrity. Low integrity software is not the correct choice to protect high integrity data. Instead, high integrity software, produced in a controlled environment, with trusted programmers, is needed, or high assurance software, evaluated by a third party. To reduce the risk that high integrity data is maliciously modified in multilevel systems, a high assurance policy enforcement mechanism, such as a security kernel, is required. It is important to include the trustworthiness of personnel into security considerations.

8 Record system actions, to detect and document malicious actions

IV. SCENARIO DESCRIPTION

This chapter introduces the reader to the virtual world of the scenario definition file (SDF) created for this analysis. It describes the setting, the policy, assets, and users of the SDF. It demonstrates how the educational goals of Chapter III are implemented, and that, indeed, this SDF generates a scenario that is simultaneously a playable game and educational tool.

A. SETTING

This section will familiarize the reader with the world the game player will immerse in while playing the scenario developed with this thesis. It contains a narrative describing the scenario's educational goal, the setting and the environment the player will face within the game. The narrative will be attached to the game, serving both the casual player and an instructor, where the latter may be looking for specific educational goals. It provides a brief overview on what to expect, if choosing this scenario. This section also contains an introductory statement to the player that will be displayed on the scenario briefing-screen of the game. It will briefly inform the player of his role in the game play, the context, the main users and most important, the tasks he will have to fulfill in order to win the game.

1. Narrative

This scenario focuses on the integrity aspect of computer security. The primary concern is the protection of stored data against unintended, malicious modification within a networked military-like environment.

The environment is the Combat Information Center (CIC) of a modern battleship. At its core is a suitable interconnection of IT components allowing the users to perform specific tasks, and an integral part of the ship's fighting abilities.

The player will assume the role of the ship's IT Officer and will have to create the CIC's IT topology from a blank environment.

Company MODERN SHIPS INC. is an enterprise working in the ship building industry. Formerly mainly concerned with merchant ships, the recent tense market

situation has led the management to leave the traditional sector and offer a new idea. Their goal is to revolutionize the military ship building market: the concept is to build a sophisticated battleship without the time consuming DOD constraints and offer the finished modern product to the military much faster than via traditional acquisition channels. In this model, the military leases the ship and pays for the offered service. The company hires the crew, mostly former military personnel, and operates the ship. It is an outsourced ship.

The first commissioned ship is called ARES. It exceeds the state of the art in battleship building. The ARES has brand new, highly sophisticated and automated sensors and effectors. To allow for a high degree of automation and the optimal use of the ship's fighting power, the ARES will be provided with the most modern hardware and software to be integrated into a highly capable and automated, hence fast and efficient Combat System. It is the player's responsibility, in the role of the ship's IT Officer, to buy the actual components the ship's crew needs to perform their tasks.

Built by US-companies, owned by international stockholders, the ARES is rented as a mercenary to organizations, primarily the US military. The ARES' goal is to succeed in every assigned mission, to quickly perform the preparatory missions with outstanding results and become ready for combat assignments, in order to increase the building company's income and convince the stockholders to invest in more ships of that type. Therefore, the ARES will be assigned a sequence of missions, starting from low-level basic training, to patrol and combat missions. Thus, it will acquire the status 'combat ready'. This will enable the ARES to be assigned to increasingly complex and important missions. These missions mean increased revenues for the company and therefore better funding for the ship.

In addition to the ship's crew, one representative of the company is stationed aboard the ship all times. Since it is the first ship of its kind and type, the representative's task is to:

a. Assist in marketing the ship, i.e. schedule port visits, receptions, presentations and VIP demonstrations.

b. Collect performance data of the ship's systems and of the crew's performance and produce a monthly Performance Report.

The graphical setting of the game is the heart of the ship: the CIC including all ship steering capabilities. The CIC is subdivided into four rooms: the main CIC, the electronic warfare room (EW Room), the navigational room (Nav Room), and the company room. Each of these rooms is a zone with different users, assets and security needs. It is one of the player's tasks, to set appropriate security settings for each zone.

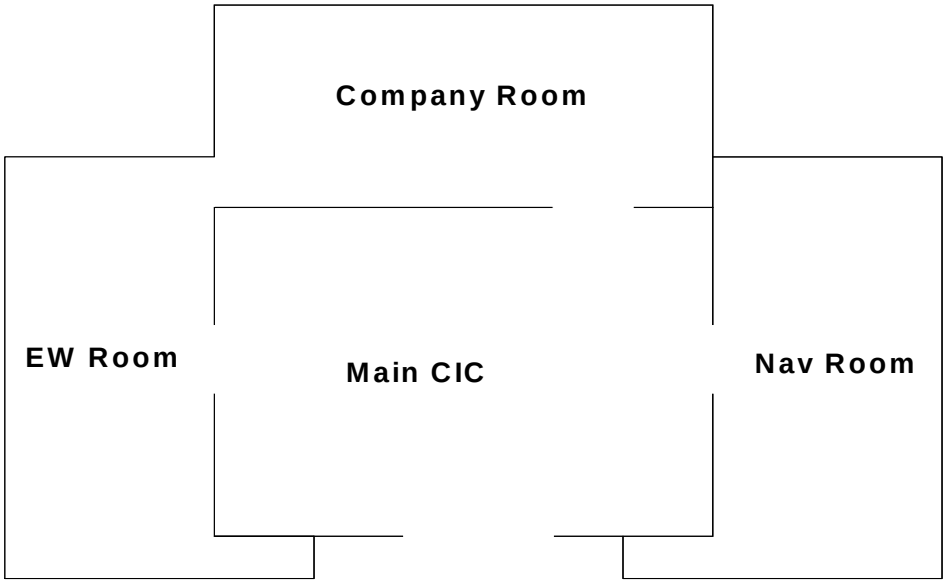

Figure 8. Room / Zone Overview Of The Combat Information Center

The player is the IT-Officer in charge of all IT. His first task is to buy components and software to allow the ship's crew interaction with the ship's systems, with each other and external sources (company), as necessary. Users need to operate on the assets they are assigned. They need to have access to these assets, they need to communicate, to pass messages and share information where required. The goal is to buy and connect components to build a Combat System (CS) that interacts with the surrounding environment at a suitable level of automation and creates a structure that allows the users to do their work, that is, to reach their goals, effectively and efficiently. The second task is to ensure security, by implementing the security policy.

The player's task is to balance and mediate between the needs of the users and constraints derived from the security policy.

His detailed tasks are to:

- Read the information about users, assets and user goals, to determine the user's needs and the protection needed by the assets.
- Buy the required components, software, security personnel, and IT staff.
- Assign the assets to the components and connect the components to networks, where necessary, to enable the users to reach their goals.
- Set the appropriate security settings, to enforce the security policy.
- Make adjustments as necessary, to keep the users happy and productive

The scenario models a training mission that runs over a period of 60 days. The player has an initial budget at his disposal, and receives a monthly budget, to perform his tasks. The ship will gain or lose money, depending on the player's choices. Unhappy or unproductive users will cause money loss, as well as security breaches. The player will win the scenario, if he does not lose all his money during the 60 days of game play. However, he will lose the game, if he cannot preserve the integrity of the high integrity assets.

2. Introductory Statement To The Player: Player Brief

The following statement is presented to the player at the start of the game:

Welcome aboard the ARES.

You have just been hired as IT-Officer for the first ship of the most modern class of battle ships. You are in charge of all IT on board this ship.

Built by MODERN SHIPS INC., owned by international stockholders the ARES is rented as a mercenary to the US Navy. Her goal is to quickly become ready for combat assignments. This will increase the building company's revenues and convince the stockholders to invest in more ships of this type. To achieve the first goal, the ARES needs to perform and succeed in a basic training mission. Achieving

the status ‘combat ready’ is a major step and will make the ARES available for complex, more challenging and better-paid missions.

The ARES has a crew of six sailors employed by the company to operate the ship. They form a team in which everyone has a special task and hence, individual needs and demands for IT-support. The company keeps a representative aboard the ship, Alice. Her task is to assist in marketing the ship, and to collect performance data of the ship’s systems and crew to produce a monthly Performance Report. Alice too, will have specific demands for IT support.

The ARES’ most valuable assets: Doctrines, Weapons, Electronic Warfare Signature, and Navigational data. It is crucial for the ship’s safety and the company’s financial well being, that the integrity of this data is preserved.

Your task is to design and maintain the ship’s network, keeping the users productive while enforcing the ship’s security policy, the main focus being to ensure the integrity of the ship’s most valuable assets.

You are given an initial budget, and you will receive a monthly budget to buy components, software, security personnel, and IT staff.

Good Luck! (See the "Game" tab for additional help)

B. POLICY

Regarding the term ‘Policy’, this paper refers to a structure that comprises an institution’s or company’s management high-level guidelines concerning information or computer security, and several derived documents, providing more specific and task-tailored directives. These guidelines are to outline the overall goals of a company, its assets, its presumed risks and its objective, possibly also stating its ‘plan of attack’, to be achieved, implementing information security.

This paper will use the definition of a security policy as provided by the encyclopedia to the CyberCIEGE game [Rivermind2 2004]:

> Computer systems (including networks of computers) can only be said to be “secure” with respect to some defined “information security policy”.

A security policy is a set of laws, rules, and practices that regulate how an enterprise manages, protects, and distributes sensitive information. The sensitivity of information has historically been categorized in terms of three different policy goals: confidentiality (…), *integrity (…), and* availability (…).

Security policies that protect the confidentiality or the integrity of information are further categorized by the basis for determining sensitivity of the information and the related constraints that should be placed on user access to the information. Three different categories of security policies are:

Mandatory Access Control Policy

Typically a management directive that identifies the sensitivities of information and the constraints placed on people who might have access to the information. Access is not granted based on the discretion of individual users. These "MAC" policies are both global and persistent. Example uses of MAC policies are protection of highly proprietary secrets from potential competitors and ensuring that only authorized accountants can alter specific critical financial data.

Discretionary Access Control Policy

Individual users or groups of users can own or otherwise control the access to information and potentially the dissemination of rights to grant access to other users. Access decisions are based on the discretion of users (often within the context of management mandates intended to constrain a user's decision to grant access based on a "need to know").

Application Security Policy

Policies defined in terms of services or the intend behavior of an application. Examples include the desired behavior of web servers; the desire to detect and strip possibly malicious attachments from e-mail; and the use of firewalls to constrain inbound traffic to web requests.

Additionally, "supporting policies" are often defined in conjunction with the above categories of policies. These supporting policies include "identification and authentication" (i.e., allowing the protection mechanisms to know who the user is), and "audit", (i.e., to enable individual accountability).

A given enterprise can have a number of security policies having differing goals (e.g., availability, secrecy and integrity). And some policies can be more precisely defined than others. And, most importantly, the consequences of violating some policies are massively more severe than the consequences of violating other policies.

The scenario developed for this paper includes both, mandatory and discretionary policies.

1. Mandatory Policy

For this scenario, four secrecy and six integrity labels are defined for Mandatory Access Control.

a. Secrecy Classification Levels

The classification levels used for modeling secrecy in this scenario are SECRET (S), CONFIDENTIAL (C), OFFICIAL USE ONLY (OUO) and UNCLASS (U).

(1) Secret (S)

> Secret is the classification level applied to information whose unauthorized disclosure could reasonably be expected to cause serious damage to the national security. Examples include information whose unauthorized release could result in the disruption of foreign relations significantly affecting the national security; the significant impairment of a program or policy directly related to the national security; the disclosure of significant military plans or intelligence operations; and the disclosure of scientific or technological developments relating to national security. [DON 1999]

The ARES operates for the US Navy to perform combat missions. Consequently, ARES possesses highly sensitive data, e.g. Electronic Warfare Signatures and a Weapon Database, important to the national security, classified as Secret. Only personnel cleared for Secret are allowed access to such information.

Damage if compromised: This asset is worth US $200 to attackers, and US $6000 to the military.

(2) Confidential (C)

> "Confidential is the classification level applied to information whose unauthorized disclosure could reasonably be expected to cause damage to the national security. Examples include information whose unauthorized release could result in disclosure of ground, air, and naval forces (e.g., force levels and force dispositions); or disclosure of performance characteristics, such as design, test, and production data of U.S. munitions and weapon systems." [DON 1999]

The ARES is a battle ship built and owned by the MODERN SHIPS INC. consortium. It is assigned to the US Navy to perform combat missions. The ARES is therefore trusted to operate with US Navy data (e.g. Tactical Picture) with a classification level of Confidential. Only personnel cleared for Confidential are allowed access to such information.

Damage if compromised: This asset is worth US $100 to attackers, and US $4500 to the military.

(3) Official Use Only (OUO). As a battle ship working for the US Navy, the ARES handles information about new technology, operational procedures or mission plans. This type of information might help adversaries to better understand US Navy maneuvers or operations and might help adversaries to catch up with technology or interfere with mission plans. Only personnel cleared for OUO are allowed access to such information.

Damage if compromised: This asset is worth US $10 to attackers, and US $100 to the military.

(4) Unclassified (U). Most information handled on the ship is unclassified. Level *Unclassified* addresses information that, if disclosed to adversaries will have no negative effect to the ships safety, or the safety of the force. Everybody is allowed access to such information.

Damage if compromised: None.

b. Integrity Classification Levels

The classification levels used for modeling integrity in this scenario are HIGH INTEGRITY (HI), HIGH INTEGRITY NAVIGATIONAL (HI Nav), HIGH INTEGRITY WEAPONS (HI WP), HIGH INTEGRITY ELECTRONIC WARFARE (HI EW), MEDIUM INTEGRITY (MI) and LOW INTEGRITY (LI). Nav, WP, and EW are used as compartments, indicating sensitivity of information of the same degree, HIGH, but separated.

(1) High Integrity (HI). Battleship ARES handles information vital to the ships' and the forces' safety, e.g. Doctrines and Weapon's database, or Navigational data. It is very important for this data to be accurate and to be protected from tampering with. Only personnel cleared for HI shall be able to modify this type

of data. Note, however, that people without clearance for High Integrity Weapons are allowed to read this type of data.

Damage if compromised: This asset is worth US $550 to attackers, and US $300,000 to the military.

(2) High Integrity Navigational (HI Nav). Navigational data, e.g. electronic maps, is vital to the ships' safety. It is very important for this data to be accurate and to be protected from tampering. Only personnel cleared for HI Nav shall be able to modify this type of data. Note, however, that people without clearance for High Integrity Navigational are allowed to read this type of data.

Damage if compromised: This asset is worth US $550 to attackers, and US $300,000 to the military.

(3) High Integrity Weapons (HI WP). Battleship ARES handles some information, mainly related to weapon systems, which is vital to the ships' and the forces' safety, e.g. Doctrines and Weapon's database. It is very important for this data to be accurate and to be protected from tampering. Only personnel cleared for HI WP shall be able to modify this type of data. Note, however, that people without clearance for High Integrity Weapons are allowed to read this type of data.

Damage if compromised: This asset is worth US $550 to attackers, and US $300,000 to the military.

(4) High Integrity Electronic Warfare (HI EW) Battleship ARES has an Electronic Warfare compartment that deals with sensitive information such as signatures of possible targets and countermeasures against incoming electronically assisted threats. It is very important for this data to be accurate and to be protected from tampering. Only personnel cleared for HI_EW shall be able to modify this type of data. Note, however, that people without clearance for High Integrity are allowed to read this type of data.

Damage if compromised: This asset is worth US $550 to attackers, and US $300,000 to the military.

(5) Medium Integrity (MI). To ensure ships' ability to participate and to properly perform in the assigned missions, the ARES needs access to highly accurate military and technical information, e.g. military mode GPS data. It

is important that the integrity of this data is preserved. Only personnel cleared for Medium Integrity shall be able to modify this type of data. Note, however, that people without clearance for Medium Integrity are allowed to read this type of data.

Damage if compromised: This asset is worth US $40 to attackers, and US $2500 to the military.

(6) Low Integrity (LI). Most of the data handled aboard ARES poses no threat to the ships' or mission safety if some of it is found to be inaccurate. Flaws, e.g. in the Daily Status Report, will easily be detected or would have no harmful effect. There are no restrictions on modifications of this type of data with the exception of any discretionary policy that might be in place.

Damage if compromised: None

Following table provides an overview of the security labels used in the SDF.

Name	Level	Category	Value to Attacker	Value to Military
Integrity				
HI	3	None	500	300000
HI EW	3	1	550	300000
HI WP	3	2	550	300000
HI Nav	3	3	550	300000
MI	2	None	40	2500
LI	1	None	0	0
Secrecy				
S	4	None	200	6000
C	3	None	100	4500
OUO	2	None	10	100
U	1	None	0	0

Table 1. Security Labels

2. Discretionary (DAC) Policy

For this scenario, the users are separated into three different groups: COMPANY, SHIP and CO. Each group is handling information that is supposed to be accessed only by members of the specified group. However, members of each group will have a motive to try and access data restricted to other groups.

a. Company

Company MODERN SHIPS INC. keeps one representative, Alice, aboard the ship. Since it is the first ship of its kind and type, Alice's task is to:

1. Market the ship, i.e. schedule receptions, presentations and VIP demonstrations.

2. Collect performance data of the ship's systems and of the crew's performance and produce a monthly Performance Report.

The ship's crew is paid and possibly exchanged on the basis of these reports, since the company is interested in proving to potential customers the outstanding performance of it's new Weapon System. To avoid disruption of the crew's performance by disgruntled employees, should they learn about their expected dismissal, the company does not want the ship's crew to see this report. The report potentially also reveals natural shortcomings in some of the ship's new systems, i.e. start up problems found in most new inventions. It is in the company's interest that neither the ship's crew, nor the customers, i.e. the US Navy, knows about those. They'd rather have the bugs fixed behind the scenes, than openly admitting problems and possibly reducing their share value.

b. Ship

Being a mercenary battle ship, the ARES handles information that might help competitors to catch up with its technology, copy efficient operational procedures, poach outstanding personnel or interfere with mission or business plans. Each time the ARES is part of a force, it has to share information with other ships of that force, e.g. the tactical picture and the ship's status (amount of fuel, ammunition, supplies, operational status of the weapon systems, defects, etc). However, the ship's commanding officer wants to limit and sanitize the information shared with the force, to make the ship's performance look better. In addition, knowing about the performance-related paycheck and bonus system of the company, the ship's crew does not want the management to find out about their shortcomings. Thus, only crewmembers shall have access labeled as SHIP.

c. Commanding Officer (CO)

The commanding officer is responsible for all actions of the ship and within the ship. It is his job to form his crew into an efficient team ready to face all the challenges ahead. He is the mediator between the military necessities, the company's wishes and the safety and well being of his crew. He ordered some data to be accessible only by the CO, i.e. some details of the next -possibly dangerous- mission, possible extensions of the ships deployment, planned crewmember exchanges. It is at his discretion, to disclose this data or parts of it at a suitable moment and to whom. Of course, any of this information is very tempting to the rest of the ship's crew. Unintended disclosure might cause disturbance between team members and might lead to a drop in performance.

C. ASSETS

Assets constitute the core of any company. They can be secret formulas, military secrets, a customer database, and any information concerning a new product. Disclosure of this information would enable competitors to reduce their technology gap, thus causing financial losses to the inventors. However, often, disclosure of an asset is not a security risk, nor the main threat. In many cases, preserving the integrity of data is more vital. Customer banking accounts, a database containing the coordinates of military targets, the website representing a company, containing links to establish communication with customers, the software of medical, military and aviation equipment, all need to be protected against malicious or accidental modification. [Irvine1 2003]

Proper protection against attacks on confidentiality and integrity of the high valuable assets is vital to succeed in the CyberCIEGE game - as it is in the real world. A detailed description of the role assets play in the CyberCIEGE game is provided in the game's encyclopedia.

Several assets of varying importance are defined for this scenario and are describe below.

1. Doctrines (DoC)

Description: Doctrines are rules consisting of Triggers, Conditions and Actions, where each is formed out of one or several units concatenated by logical operators. Doctrines pre-set the ‘Operational Parameters’ of both, sensor and effector’s interfaces. They are programmed by a special team of experienced and trusted personnel and allow for a set of predefined parameter settings to adjust the ship’s Systems to pre-defined states, e.g. SAFE, NORMAL and CRITICAL. SAFE means all automated weapon engagements are blocked. NORMAL means that the ship is preset for routine peace operations, allowing automated engagements only on simulated foes. CRITICAL means the ships systems are prepared for automated weapon engagement. Other states can be generated at the commanding officer’s discretion.

The Doctrines are of high relevance to the ship’s precise performance and to its safety. It is vital to preserve their integrity. If the data is corrupted, the system settings will be in an undefined state. This will most likely lead to misfire or automated engagement of any contact, possibly friendly or neutral. Any of these occurrences either endangers the ARES’ survival in combat or, due to friendly or neutral casualties, leads to dismissal of the ship from US Naval service. The resulting bad publicity would lead to financial bankruptcy of MODERN SHIPS INC.

Although this singular set of rules applies to a specific equipment and environment, the data is classified as secret, because information about the ship’s weapon systems can be deduced from it.

For the ARES, the Doctrines are its most valuable asset.

Classification: S, HI WP

Discretionary Controls: Ship.

Asset Goals: Keep the Doctrines current and use them to set the weapon system to the most effective configuration for the current environment. To modify the asset, special software is required: Software of type DEFENSE 4T. Use the key "e", to look up details in the encyclopedia.

2. Weapon Parameter Data Base (WPDB)

Description: The Weapon Parameter Data Base contains specific information about the ship's weapons. This includes the ammunition needed, the weapon range, the pre-sets for ammunition and weapon-specific variables for certain shoot or drop conditions, e.g., the distance, at which a missile is supposed to activate its radar, the search window for the missile's search sensors, the upper and lower search depth for a torpedo, and the minimal distance for a weapon to initialize it's warhead. Another very important data set is the alignment variables of the weapon and fire control system. They ensure that the weapon really points towards the selected target.

If this data is corrupted, weapons might be fired at a target different from the one selected, thus posing the threat of miss-fire, friendly fire or engaging a neutral contact. A flaw in the warhead initialization parameters might lead to an early explosion, threatening the ship's safety.

Any of these occurrences will be considered as total failure of the weapon system. The ARES will be decommissioned, and the company will face bankruptcy.

Since adversaries might be able to better predict the ARES' capabilities, only personnel cleared for secret shall have access to this information.

Classification: S, HI WP

Discretionary Controls: Ship

Asset Goal: Keep the Weapon Parameter Data Base current. The data is used to update the ship's weapon systems, to allow for fast and correct use of the weapons. To modify the asset, software of type DEFENSE RAT is required. Use the key "e", to look up details in the encyclopedia.

3. Electronic Warfare Signatures (EWS)

Description: Electronic Warfare Signatures are a digital fingerprint of electromagnetic emissions of a radar or communication system. EWS are used to classify and identify intercepted electromagnetic signatures. By comparing the intercepted data to the repository, the system can identify hostile ships and weapon systems. It can identify incoming hostile missiles and provide necessary data for

intercepting, e.g., height, flight profile, and speed. Should the EWS be modified, the ship could not properly identify a foe, and could not effectively counter incoming weapon systems. Serious damage or even total loss of the crew and the ship would be the consequence.

To protect the EWS, only specially trained and cleared personnel are supposed to update the database.

Classification: S, HI EW

Discretionary Controls: Ship.

Asset Goals: Keep the EWS current and use them to asses the type and threat of intercepted signals. To modify the asset, special software is required: software of type DEFENSE RAT. Use the key "e", to look up details in the encyclopedia.

4. Daily Status Report (DSR)

Description: The Daily Status Report (DSR) provides an overview of the current status of the ship's systems to the CO, e.g., the weapon, or propulsion systems. Each department has to contribute information concerning its systems. The DSR is a means for the CO to assess the current capabilities of the ship, based on system availability and status. The DSR is for official use only, and not to be disclosed to non-ship members.

Classification: OUO, LI

Discretionary Controls: SHIP.

Asset Goals:

Write DSR: Add the current status of the department’s systems to the DSR, to keep the CO informed about the ship’s capabilities and problems. The information helps the CO make sound decisions. Software of type SPREADSHEET is required to perform the task.

Read DSR: Read the current status of the ship’s systems to get information about the ship’s capabilities and problems. Software of type SPREADSHEET is required to perform the task.

5. Performance Report (PR)

Description: One of the company representative's tasks is to collect performance data of the ship's systems and of the crew's performance and produce a monthly Performance Report. The technical manager of the company utilizes this report to prioritize research, production and repairs. The marketing manager has developed a plan to influence the crew's performance by incentives. Employees showing high performance will receive a higher salary and possibly bonus payments. Employees with a performance below a certain threshold over a specified period of time will face dismissal.

The CO is allowed to read the report before submission. However, he must not alter the original contents.

Classification: OUO, LI

Discretionary Controls: COMPANY, CO

Asset Goals: Monitor the performance of the ship's crew and weapon systems, and make a monthly report for the management. The technical and marketing managers need the report to prioritize research, and to streamline the incentives plan. To modify the asset, software of type WORD PROCESSOR is required. Use the key "e", to look up details in the encyclopedia.

6. Frequency Plan (FP)

Description: The ARES is rented by the US Navy and participates in exercises, maneuvers and missions consisting of several ships. To establish communication with these units several communication lines have to be established, using the frequencies assigned in the Frequency Plan (FP). The FP describes which frequency has to be used for all the ship's communications, e.g., for the tactical communication on the Anti Submarine Warfare (AAW) line, during an AAW exercise, or for the ship-to-ship safety line during close formation exercises.

Classification: OUO, LI

Discretionary Controls: SHIP

Asset Goals: Keep the Frequency Plan current, so that the ship's crew is able to communicate with each other, the shore facilities and the other ships of the force. To modify the asset, software of type SPREADSHEET is required. Use the key "e", to look up details in the encyclopedia.

7. Email (EM)

Description: The email system on board the ARES is important to every sailor. As a morale measure, to ensure the sailors' connectivity to news, their family and friends, at least one terminal has to be provided for email traffic. At a minimum they have to be fitted with Internet access, a mail messaging functionality (create, send and receive) and a picture viewer. It is the CO's discretion to allow or interrupt access at certain times, according to the threat level in force.

Classification: U, LI

Discretionary Controls: Ship, COMPANY

Asset Goals: Every person on board the ARES shall have access to the email system. It is, however, the CO's discretion to restrict access at certain times, according to the threat level in force. To modify the asset, software of type EMAIL CLIENT is required. Use the key "e", to look up details in the encyclopedia.

8. Internet Web Page (WP)

Description: MODERN SHIPS INC. utilizes a webpage to present information about their products to potential customers. Unclassified information about the ARES' structure, capabilities, past missions, pictures, port visits and times to visit the ship is posted. The Web Page is stored on a company server on shore. Alice, the company representative on board the ARES is keeping the information current. The information presented by the company is, in many cases quite idealized, especially concerning the workload and the payment structure. Some of the crewmembers do not like Alice and disagree. Given the chance, they would modify the page to get the representative into trouble.

Classification: U, MI

Discretionary Control: COMPANY

Asset Goals: Description: Create and update the company’s Web Page with current data about the ARES. To modify the asset, software of type Web Server is required. Use the key "e", to look up details in the encyclopedia.

9. Mission Plan (MP)

Description: The Mission Plan is created by the CO and describes the ship's current mission. It provides the relevant information for each department. This allows the crew to set their individual goals and identify their specific tasks. For example, the navigator can prepare the charts for the journey, the DOC Officer can prepare the appropriate Doctrines, and the Communications Officer can prepare the frequency plan. The information in the Mission Plan is to be accessed by the Ship's crew only.

C*lassification:* OUO, LI

Discretionary Controls: CO

Asset Goals:

Create Mission Plan: Keep the relevant information in the Mission Plan current, so that the crew knows what lies ahead and can perform their tasks. To modify the asset, software of type WORD PROCESSOR is required. Use the key "e", to look up details in the encyclopedia.

Read Mission Plan: Read and evaluate the information in the Mission Plan. You need to know what lies ahead, and derive and prepare your tasks. To read the asset, software of type WORD PROCESSOR is required. Use the key "e", to look up details in the encyclopedia.

10. Tactical Picture (TP)

Description: The Tactical Picture is the result of scanning the environment with sensors and processing this data, together with information gathered from other sources, e.g., intelligence, signature databases, etc., to obtain as much information about a contact as possible. The goal is to know about every asset in the area of interest, to be able to classify them as neutral, friend, or foe, and to track their movements. Modern systems should allow for sharing this information with other (friendly) units, if the ship is assigned to a task force, to generate more accurate

information by combining the capabilities of different sensors and to extend the area of coverage, exceeding the limited range of one’s own sensors.

This information is important to allow for an accurate assessment of the situation, and for appropriate measures, e.g., to counter a threat. However, it is valid for a short time only, and it is constantly updated. Furthermore, transmission integrity is of minor concern, since the data exchange must be performed over secure (protected) lines only, to prevent adversaries from observing it.

Classification: C, LI

Discretionary Controls: SHIP

Asset Goals:

Create Tactical Picture: Create the Tactical Picture in the ARES' operational area. That is, obtain the necessary information about all contacts and classify them as neutral, friend, or foe. To modify the asset, software of type MANAGEMENT is required, and access to the Navigational Data is necessary. Use the key "e", to look up details in the encyclopedia.

Read Tactical Picture: Read the Tactical Picture to obtain a current and accurate overview of the tactical situation in the ARES' operational area. Be informed about all contacts and their status: neutral, friend, or foe. To read the asset, software of type MANAGEMENT is required. Use the key "e", to look up details in the encyclopedia.

11. Navigational Data (Nav Data)

Description: Navigational Data comprises electronic maps, GPS position fixes, data about wind, weather, water drift, the ship’s position, course, speed, etc. This data is required to maneuver the ship correctly and to allow the CO to keep the ship safe. If the data is incorrect, the ship is in great danger to collide with obstacles, or run aground.

Classification: OUO, HI Nav

Discretionary Controls: SHIP

Asset Goals: Use all your skills and navigational means (electronic maps, wind and weather data, GPS) to calculate ARES' accurate position, course, speed, and the surrounding conditions: wind, depth, obstacles, tide, etc. To modify this data, software of type DEFENSE 4T is required. Use the key "e", to look up details in the encyclopedia.

The following table provides an overview of the assets used in the SDF.

Name	**Abbreviation**	**Integrity Label**	**Secrecy Label**	**DAC (read/write/control/execute) Y: yes; N: no; X: whichever**
Electronic Warfare Signatures	EWS	HI EW	S	Ship YNNY
Doctrines	DoC	HI WP	S	Ship YNNY
Mission Plan	MP	LI	OUO	Ship YNNN
Daily Status Report	DSR	LI	OUO	Ship YYNN
Email	EM	LI	OUO	Ship YYYY
Web Page	WP	MI	U	PublicYNNN
Performance Report	PR	LI	OUO	Company YNXX Alice YYXX
Weapon Parameter Database	WPDB	HI WP	S	Ship YNNY
Frequency Plan	FP	LI	OUO	Uhura YYYY
Tactical Picture	TP	LI	C	Ship YYYY
Navigational Data	Nav Data	HI Nav	OUO	Ship YNNY

Table 2. Overview Of Assets

The following table provides an overview of the asset goals used in the SDF.

Name	**Assests**	**AccessMode (read/write/control/execute) Y: yes; N: no; X: whichever**	**user**	**Comments**
Create Mission Plan	Mission Plan	YYXX	James	SoftwareType: WORD PROCESSOR
Read Mission Plan	Mission Plan	YXXX	all except: Alice, James	SoftwareType: WORD PROCESSOR
Modify EWS	EWS	YYXX	Wesley	SoftwareType: DEFENSE 4T
Modify DoC	DoC	YYXX	Leonard	SoftwareType: DEFENSE 4T
Write DSR	DSR	YYXX	all except: Alice, James	SoftwareType: SPREADSHEET
Read DSR	DSR	YXXX	James	SoftwareType: SPREADSHEET
Modify Performance Report	Performance Report	YYXX	Alice	SoftwareType: WORD PROCESSOR
Modify Weapon Database	WPDB	YYXX	Leonard	SoftwareType: DEFENSE RAT
Modify Frequency Plan	Frequency Plan	YYXX	Uhura	SoftwareType: SPREADSHEET
Create Tactical Picture	Tactical Picture Navigational Data	YYXX YXXX	Joe	SoftwareType: MANAGEMENT
Read Tactical Picture	Tactical Picture	YNXX	James	SoftwareType: MANAGEMENT
Modify Nav Data	Navigational Data	YYXX	Pavel	SoftwareType: DEFENSE 4T
Collect Reporting Data	EWS	YXXX	Wesley	SoftwareType: REPORTING
Update Web Page	Web Page	YYXX	Alice	Web Server
Modify EW Picture	EW Picture	YYXX	Wesley	SoftwareType: MANAGEMENT
Read Email	Email	YYXX	all	SoftwareType: EMAIL CLIENT

Table 3. Asset Goals

D. USERS

Each CyberCIEGE SDF has a set of users who work for the enterprise, on specific tasks and, by achieving their goals, earn money for the enterprise. The enterprise of this thesis' SDF is the ARES. The users working in the virtual world of the ARES' CIC, are subdivided into two groups; the ship's crew and company representatives. This paragraph describes each of the SDF's users.

1. Ship's Users

There are several members belonging to the ship's crew.

a. Commanding Officer (CO)

Name: James T Maury.

Cleared to Secret, LI.

Belongs to groups CO and SHIP.

Description: James is the CO of the ARES. He is around 40 years old and has a wife, a son and a daughter. This ship is his third CO assignment. He held command on two modern frigates of the US Navy, before he was offered the chance to become CO on the ARES, the most modern ship available. James is an experienced sailor, knows Navy tactics and procedures and has proven to be a reliable and battle proof officer. As CO, James is responsible for all actions of the ship and within the ship. It is his job to form his crew into an efficient team ready to face all the challenges ahead. He is the mediator between military necessity, the company's wishes and the safety and well being of his crew. James is a cheerful person; he loves new technology, although he is not well trained, nor skillful with IT tools. As a former Naval Officer, he knows that, for the military, security is very important. In order to keep his crew informed and focused on the tasks ahead, James needs to write the Mission Plan and update it on a regular basis. At all times, James needs to have read access to the Daily Status Report, to be current on the operational capabilities of his ship and to set priorities. Furthermore, James wants to read the company's Performance Report, to monitor the information exchange with the company. On station, he must have access to the Tactical Picture, and the Navigational Data, to analyze the situation and prioritize actions.

b. Warfare Coordinating Officer (WCO)

Name: Joe Strain.

Cleared to Confidential, LI.

Belongs to group SHIP.

Description: Joe is a well-educated and skillful Naval Officer. He has served on several Frigates and established his career in the Operations Department and worked on becoming a commanding officer. Joe is a cheerful, reliable person, and,

although moderately educated in IT technology, he is willing and able to learn quickly. Joe's main responsibility is to compose the Tactical Picture, and keep it current. To prepare for the tasks ahead, he needs to frequently read the Mission Plan. Daily, he is required to write his portion of the Daily Status Report.

c. Electronic Warfare (EW) Operator

Name: Wesley Truxtum.

Cleared to Confidential, HI EW.

Belongs to group SHIP.

Description: Wesley is leading the ARES' electronic warfare. He uses the modern electronic warfare system to detect, analyze, classify electronic emissions within the ARES' operational area and correlate them to detected tracks. In a combat situation, Wesley operates the active electronic countermeasures, e.g. generates false targets, or generates emissions that interfere with or possibly destroy the adversary's sensors and weapon systems. Wesley is an excellent soldier, very reliable, and a hard worker. He gathered a lot of experience with the EW systems at previous assignments and has attended introductory IT courses. In addition to his main task, modifying the EW Signatures, Wesley needs to read the Mission Plan once a day and to prepare his department's portion of the DSR.

Later in the game, Wesley will be asked to collect Reporting Data of his system.

d. Navigator

Name: Pavel Codazzi.

Cleared to Confidential, HI Nav

Belongs to group SHIP.

Description: Pavel is a cheerful person and hard worker. He has been well trained in the newest IT equipment of the ARES, as preparation of his role as the ship's navigator. He displays lots of good will, he is very helpful and he tries hard to integrate into the ship's crew. Unfortunately, the records from his previous command are missing as of today. Pavel's main task aboard the ARES is to generate accurate

Navigational Data and keep it current at all times. The ship's safety relies on this work. In addition to this task, Pavel needs to read the Mission Plan daily, and to prepare the portion of the DSR concerning the navigational system.

e. Communications Operator (ComsOp)

Name: Uhura Winthrop.

Cleared to Classified, LI

Belongs to group SHIP.

Description: Uhura is a young, cheerful person, who recently entered the service. She had only one assignment before switching to ARES, which was as a communications operator, however, it was on an old ship, with older systems. Uhura knows all the naval communications procedures, but has not much experience with new IT equipment, yet. Uhura's main task is to prepare the ship's Frequency Plan and keep it current. To work effectively, she needs to read the Mission Plan at least once a day. Uhura is responsible for the portion of the DSR, concerning all communications equipment.

f. Doctrines Operator (DocOp)

Name: Leonard Boggs.

Cleared to Secret, HI WP.

Belongs to group SHIP.

Description: Leonard is an experienced, reliable, and responsible soldier, who worked in the weapons department of several ships, before he was asked to join the ARES' crew. He is always cheerful, friendly, and very helpful. Although he had no previous experience, with Doctrines, he is well trained in everything related to this new tool. However, he is only moderately trained in general IT equipment, but eager to learn. Leonard's main task is to keep the Doctrines and the Weapon Database current. His two additional, but minor tasks are to read the Mission Plan, and to contribute to the DSR. However, this requires only a small portion of his time.

2. Company Representatives (CR)

There is one company representative stationed permanently aboard the ARES.

Company Representative (CR)

Name: Alice Fitzroy.

Cleared to Official Use Only, MI.

Belongs to group COMPANY.

Description: Alice is the company representative on board ARES. Alice is an intelligent, good looking and ambitious person, with detailed computer skills. She is very loyal to her company, enjoys being on the ARES and likes James, the CO. Alice understands the necessity of a security policy, however, she is quite a nosy person, and due to her task, she really would like to know in advance the ship's plans, i.e. the Mission Plan. Alice' primary task is to manage the marketing of the company's newest and best product: the new ship, the ARES. She has to plan and schedule receptions in the harbors to be visited, perform presentations and invite VIP's to demonstration tours. She is responsible for the production of the company's Web Page about the ship and to keep the information current and accurate. Alice' second task aboard the ARES is to prepare the Performance Report concerning the ship's crew and system performance for her company.

The following table provides an overview of the users modeled in the SDF.

Name	Integrity Label	Secrecy Label	Access to Assets mode: r - read m - modify
James T Maury	LI	S	MP m DSR r TP r
Wesley Truxtum	HI EW	S	EWS m MPr DSR r
Leonard Boggs	HI WP	S	DOC WPDB MP DSR
Pawel Codazzi	LI	C	NavData MP DSR
Uhura Winthrop	LI	C	CC MP DSR
Joe Strain	LI	C	TP m MP r DSR r
Alice Fitzroy	MI	OUO	PR m WP m

Table 4. Users

3. Staff Members

In addition to the users described above, the SDF provides several users as members of the IT and Security staff, whom the player can employ to assist the ship's crew and the company representative. He can dismiss members of the staff, if he is not satisfied with their performance.

These staff members are:

a. Security

John Cullen: John is a former Marine. He could not cope with the difficult tasks of his corps and decided to become a security guard instead He is very

skilled and well trained. John is always in a good mood, however no information is available about his past.

Winsome Adam: Winsome is a former Marine, well trained in weapons and Marshall Arts. She is very interested in IT technology and makes a perfect guard.

Justin Talbot: Justin is a former Marine, with excellent training. He is the perfect guard. However, he is a little chauvinistic, but takes psychological training, to get better.

Dave Wilson: Dave is a newly trained security guard, who only recently finished his training. He is said to be moderately skilled, but has a high potential for improvement.

Debby Smith: Debby is well trained in weapons and Marshall Arts. She is very interested in IT technology and makes a perfect guard.

The following table provides an overview of the security personnel.

Name	Trustworthiness	Initial Training	Skill	Happiness	Cost US $
John Cullen	60	90	100	90	500
Winsome Adam	100	85	90	95	800
Justin Talbot	100	100	100	90	1000
Dave Wilson	90	70	80	90	650
Debby Smith	100	85	90	95	800

Table 5. Security Guards

b. IT Support

JD Smith: JD is a CS graduate. He loves to work in the IT field and specialized in IT Security. He is very capable, effective and reliable. He is worth the high salary.

George D. Jones: George is a former mathematician and IT guy without standards. He lost his job but seems to be willing to try again. Not much data is available about his past.

Naomi Joule: Naomi is computer geek. She is very friendly, always helpful and always has a smile on her lips. Naomi had some trouble with "computer crimes" before she joined the Navy. However, the Navy performed a thorough background check and cleared her for confidential. Naomi is very productive and certainly worth the salary.

Buddy Martin: Buddy is a computer geek. He is trustworthy and well trained. Buddy is a friendly and good worker, who, however, gets distracted sometimes.

Chloe Steele: Chloe is a computer geek. She is very friendly, always helpful and always has a smile on her lips. Chloe is very productive and certainly worth the salary.

The following table provides an overview of the IT Staff.

Name	Trustworthiness	Initial Training	Skill	Happiness	Cost US $
JD Smith	100	100	100	100	3000
George D. Jones	60	60	60	60	1000
Naomi Joule	80	80	99	90	2000
Buddy Martin	90	80	80	90	1500
Chloe Steele	80	80	99	90	2000

Table 6. IT Staff

The player has to accommodate the needs of eight users, seven soldiers, and one company representative. To assist these users in IT matters, he can employ IT staff, from a set of four persons with different skills. The player can choose among four security guards of different backgrounds and skills.

E. IMPLEMENTATION OF EDUCATIONAL GOALS

The above sections introduced the reader to the virtual world of the scenario definition file (SDF) created by this thesis. This section describes how the SDF implements the educational goals of Chapter III. Using the above description of assets and users, it answers the first research question of the thesis: can a scenario be

developed, such that it is simultaneously a playable game and educational tool, while illustrating integrity issues in a military-like networked environment?

1. Software Integrity

The major educational objectives concerning the first two educational goals are for the player to realize the impact of software integrity on the ability to preserve the integrity of assets, that claims of vendors tend to exaggerate the security features of their products and that high integrity software, produced in a controlled environment, with trusted programmers, or high assurance software, evaluated by a third party, is needed, to operate on high integrity data.

To implement both educational goals, both high and low integrity assets, and an asset of medium integrity have been introduced. The ARES' high integrity assets are the Electronic Warfare Signatures (EWS), the Doctrines (DoC), the Weapon Database (WPDB) and the Navigational Data (Nav Data). They are classified high integrity, with different compartments. The EWS are classified HI EW, the DoC and WPDB are classified HI WP and the Nav Data is classified HI Nav. The ship's low integrity assets are the Mission Plan (MP), the Daily Status Report (DSR), Email, the Frequency Plan (FP), the Tactical Picture (TP), and the Electronic Warfare Picture (EWP). The company's assets are the Performance Report (PR), classified low integrity, and the Web Page, classified medium integrity.

The existence of assets of different integrity levels generates the requirement for the player to choose between software of different integrity, which is to be provided to the users, to operate on the assets. While low integrity software is sufficient and cost efficient, for use on low and medium integrity assets, high integrity software is needed for the high integrity assets.

The asset goals related to each asset specify the type of software an user needs to access his assets. For example, to modify the EWS, the user Wesley, is required to have software of type DEFENSE 4T at his disposal. The assets DoC and Nav Data also require software of type DEFENSE 4T, whereas the asset WPDB requires software of type DEFENSE RAT. For each type of software, there are at least two different product choices. Application 'Scare Crow Defense Systems 4T' and 'Skunk

Cellars 4T' are both of type DEFENSE 4T. Thus, both are possible choices to provide user Wesley the software necessary to access his asset EWS, and thus, reach his goal to 'Modify EWS'. However, there are differences between the applications. For example, 'Scare Crow Defense Systems 4T' has a price of US $10,000 and a moderate integrity value (400), while 'Skunk Cellars 4T' is more expensive, having a price of US $20,000, but it also is a high integrity product (900). Because the attacker value for assets classified HI EW is high (550), the cheaper software is not sufficient to preserve the integrity of the asset EWS. To access low integrity assets, low integrity applications are sufficient and are unlikely to cause serious malicious modifications, because the value of the asset to an attacker is also very low. For example, the asset MPL requires software of type WORD PROCESSOR. Applications of that type are 'WordSmyth', price US $200 and integrity value 20, and 'Word Triangle', priced at US $20 with an integrity value of 80. Despite these differences, both applications will not cause malicious modification of the data, because the value of assets classified as low integrity is of no interest for an attacker.

The description in the game's 'Software' screen and encyclopedia is used to convey to the player, how much trust he can have into an application's integrity. For example, the description provided in the game's 'Software' screen for the application 'Molo Defense Systems RAT' and 'Scare Crow Defense Systems 4T' is: "Pass all gov't tests for safety" [Rivermind2 2004]. This corresponds to low integrity values. The description offered for the applications 'Skunk Cellars RAT' and 'Skunk Cellars 4T' is: "Built and maintained in a controlled environment, with trusted programmers." [Rivermind2 2004]

The player's choice for the software offered to the user to operate on his assets and necessary to reach his goals, affects the chance to win the scenario. If the player chooses low integrity products for the ship's high integrity assets, he will not be able to preserve the asset's integrity for the required 60 days and he will lose. However, it is not possible for the player to buy applications of high integrity for all his users and assets. Therefore, playing this scenario and making the correct choices, the player will win the game and learn about the effect of software integrity to the security of high integrity assets.

2. MAC Enforcement Mechanism

The major educational objective of the third educational goal described in Chapter III is to educate the player about MAC enforcement mechanisms:

a. It is important to choose a trusted MAC enforcement mechanism that is architecturally positioned such that discretionary mechanisms cannot alter its enforcement functionality. Thus, to preserve the integrity of high integrity data in multilevel systems, which use software of various assurance levels. A security kernel is an appropriate mechanism.

b. It is important to properly implement the RM concept. A security kernel will not be able to protect high integrity data if the security implementation assigns low assurance software to access the data.

This educational goal is implemented by generating the need for an user to be given read access to a high integrity asset, while using an application of low integrity. This need leads to the requirement of a component that has a MAC enforcing mechanism, that enforces a mandatory security policy and allows the low integrity software read access, while denying it write access. For example, Wesley's goal is to modify the EWS. He requires software of type DEFENSE 4T. To preserve the assets integrity, the player needs to buy the application 'Skunk Cellars 4T', which is of high integrity. A component with a high integrity OS suffices, a MAC enforcement mechanism in not necessary.

Several days into the game, Wesley is assigned the task to collect reporting data of the EW system. To meet this task, he needs read access to the EWS with software of type 'REPORTING'. Both applications of this type, 'Scare Crow Defense Systems T&M' and 'Molo Defense T&M', are of low software integrity (values 200 and 300). The player can choose to buy Wesley one of the above applications, while still using a component without a MAC enforcement mechanism, or, the player can decide to buy a new component, which enforces a MAC policy. The first choice will lead to the asset's modification and the player will loose the game, because the low integrity application will most certainly have malicious code that will have direct access to the EWS. If the player chooses to buy a component with a MAC

enforcement mechanism, he is given the choice between components. The components have different prices and MAC enforcement mechanisms of different assurance. To illustrate, that a high assurance MAC enforcement mechanism, a security kernel (SK) is required to preserve the integrity of high integrity data, the player will loose the game, if he chooses a component with a MAC enforcement mechanism of low assurance. For example, the correct choice to enable Wesley to meet his new goal would be to buy a component with a SK evaluated at EAL 6 or EAL7, e.g., buying a component with the OS 'Green Shade Core'. Buying a component with the OS 'Trusted Populos Desktop' that has a MAC enforcement mechanism of EAL 4 will not suffice to win the game.

The need for the player to consider buying components with a MAC enforcement mechanism is further generated, by introducing the need to connect some components holding high integrity assets to a network of lower integrity. For example, the CO and Joe need to read the Navigational Data. Therefore, the player needs to connect Pavel's, Joe's and the CO's component to a network. To protect the Nav Data from unauthorized modification, Pavel's component needs to have a SK, because the CO and Jo are only cleared for low integrity. Similarly, Joe needs a component with a MAC enforcement mechanism, because he needs to access data of different sensitivity. He needs to read the Nav Data, classified LI OUO, while modifying the TP, classified LI C. Hence, a MAC enforcement mechanism of moderate assurance, EAL 4, is sufficient, because the asset TP is only of moderate value to an attacker.

Playing this scenario, the player is encouraged to consider the value and nature of MAC enforcement mechanisms. The player learns that low assurance MAC enforcement mechanisms suffice for assets of low integrity, while a SK is vital, to protect high integrity assets.

3. Social Engineering

The major educational objective of the fourth educational goal described in Chapter III is to educate the player that security measures include more than technical means. It is vital, to include the notion of trustworthiness of personnel in security considerations for the system.

This educational goal is implemented by using two parameters that affect the integrity value the game engine calculates for a user: the parameters ‘Trustworthiness’ and ‘InitialBackGroundCheck’. One user is given a low trustworthiness value. Pavel, who is responsible for modifying the high integrity asset Nav Data, is given a trustworthiness value of 01. In addition, the value for the ‘InitialBackGroundCheck’ of the clearance label HI Nav is set to ‘medium’. The combination of these two values results in successful social engineering attacks on the user, who is lured by an attacker to maliciously modify his high integrity asset; in the case of Pavel and the Nav Data. These attacks occur even if the player has chosen correct component settings, as discussed above. The player will be able to prevent a social engineering attack from being successful, if he modifies the accessible parameter, the background check. The player needs to realize the low background check value (‘Back Check Medium’ in the user screen) and to buy a high background check for the clearance level ‘HI Nav’ at the game’s security screen.

During the game, the player has to monitor his financial resources, the happiness and productivity of his users and the needs of the users. The player’s choices concerning the components and software he buys for the users, and his decisions concerning the connectivity of the components, influences the way users can access their assets and achieve their goals, which influences the parameters user happiness and productivity. These parameters are also influenced by the security measures taken to implement the security policy and by the effectiveness of these measures. Very tight security measures, video cameras, for example, and the need to change to another zone to achieve an asset goal, have a negative impact on the user happiness. Incorrectly configured components lead to security problems, such as viruses, and result in monetary penalties, as well as in a reduction in user happiness. Some asset goals, e.g., ‘Collect Reporting Data’, are introduced later within the game’s playtime. They result in the need to reconsider the network design and security measures, chosen at game start and possibly in a modification of the previous player choices.

Additionally, ‘pop-up’ messages inform the player about new user needs, low user productivity and security breaches, generating a game-user interaction throughout the game and contributing to the playable aspect of the SDF.

F. SUMMARY

This thesis’ SDF creates a virtual world situated in the CIC of a modern battle ship. It generates users with different attributes and various clearance levels, belonging to two different groups (soldiers and civilians), and having different needs. Different assets and asset goals influence these needs. The assets are of various classification levels, the main focus being the integrity aspect of security, and generate the need for careful consideration about the hardware and software the player needs to purchase, and the network connections the player needs to establish, to enable the users to reach their goals and to be productive. Specific users, assets, and asset goals generate game-play situations that can educate the player about basic integrity principles, implementing the thesis’ educational goals. These users, assets, and asset goals, in addition to other parameters, for example, user happiness, productivity, messages indicating new user needs, or attacks, allow for user-game interaction and constitute the SDF’s playability.

Consequently, the thesis’ first research question can be answered. Yes, indeed, a scenario can be developed, such that it is simultaneously a playable game and educational tool, while illustrating integrity issues in a military-like networked environment. The SDF generated by this thesis is an example.

The next chapter is concerned with answering the thesis’ second research question: Is it possible to validate that the CyberCIEGE game engine produces expected results from a specific scenario definition file?

V. TESTING

A. TEST STRATEGY

The CyberCIEGE game models security concepts in IT environments. This thesis created a Scenario Definition File (SDF) to teach a player specific security topics while playing the game. Depending on the player's choices in designing and implementing components and networks, security will be preserved or compromised.

The specific educational goals of the game scenario developed by this thesis, are modeled and implemented within a complex SDF. However, the development process uses an incremental approach, starting with small, simple scenarios and stepwise adding more complexity. Along the development path, testing is used to verify that the intended goal can be modeled within the boundaries of the Scenario Format Template (SFT) and that the game engine properly simulates real world behavior. The testing phase concentrates on whether the game engine is able to accept the settings of the SDF and generate actions one would expect in the real world, such as virus attacks in the absence of anti-virus measures (i.e., AS software, scanning of attachments, prohibition of user-introduced software) or integrity attacks in the presence of low assurance software.

The testing approach that seems to be the most suitable for this project is the Verification Validation and Accreditation (VV&A) [DMSO2 2003], more specifically, the face validation process[9], as part of VV&A. Since the game is based on a behavioral model, rather than exact physics, testing has two inherent properties. First, it is nondeterministic in nature, i.e. the dependency between cause and effect is not a function. Rather, the result of a specific action or setting is expected to occur with a certain probability. Second, one cannot apply quantitative measures, but needs to focus on qualitative measures.

[9] Definition of face validation process by [DOD 1997]: The process of determining whether a model or simulation seems reasonable to people who are knowledgeable about the system under study, based on performance. This process does not review the software code or logic, but rather reviews the inputs and outputs to ensure that they appear realistic or representative.

Reviewing DOD 5000.59-M [DOD 1997] and the current state of the game development process, which has no high level test plan and no VV&A process implementation in the development phase, face validation seems to be the best choice as a strategy for performing the above test goals. DOD 5000.59-M [DMSO1 2001] describes this process as being "useful mostly as a preliminary approach to validation in the early stages of development."

Using the face validation process and using the 'SDF – developer' as the subject matter expert, allows for several basic tests of the game engine and the developed SDF, while keeping testing costs and its time scale in proportion. This approach provides several test cases as building blocks to be incorporated in a high level, overall test strategy, allowing a continuation of the validation process and in-depth follow-on testing.

It is important to note, that the project is still in a very formative phase, causing several changes and adjustments to integral components. During the development of the SDF and during the testing phase, the game engine has been subject to software changes and fine-tuning. This has also been the case with the Scenario Template Format (SFT). This is reflected in the following chapter, as some of the test cases have been tested with more than one version of the game engine.

B. TEST CASES

This section describes some of the most significant test cases. The test cases follow a common three-fold structure. The first paragraph describes the scope of the test case, contains the name of the test-SDF and references the applied educational goal. The second paragraph depicts the expected results, while the observed results are the subject of the last paragraph.

The test cases differ in magnitude and scope. Starting with small, simple scenarios, the test cases analyze single, isolated security concepts. They culminate in complex scenarios, combining several concepts into interrelated scenarios.

The first test cases model security concepts derived from the four educational goals concerning integrity as described in Chapter II Section B of this document. It

uses very small and simple scenarios. They usually contain only one user and one asset. The intention is to isolate the educational goal from other effects that would be present in a real world scenario and verify that the game engine generates attacks specific to this simple environment. After the verification of the isolated educational goal, the scenario is gradually expanded to include more users, assets, components and secrecy and integrity issues. The final test case is the full scenario as described in Chapter IV.

Test Cases 1 through 6 use the following general setting, and differ only in a small number of test parameters.

Wesley is the only user modeled in the scenario. He is working on a single stand-alone component. Wesley's goal is to modify the single asset, the EW Signatures, which is classified High Integrity EW (HI EW). The asset's secrecy label is CONFIDENTIAL (C). Wesley has the appropriate clearance (C, HI EW), is well trained and has a high trustworthiness. The Integrity value of label HI EW is high (1000), as is its attacker value (800). The Secrecy value of label C is low (100); its attacker value, however, is 0. The component resides in a walled office within an office area patrolled by a security guard and protected by a key lock. Due to the nondeterministic nature of the test subject and setting, the expectation is to experience at least one integrity attack and no attacks on secrecy, within 60 days.

Changing to CyberCIEGE version March 11, several parameters of the game engine were modified. The attack frequency has been reduced and the importance and impact of physical security has been increased. To incorporate these changes, the SDF's for the Test Cases 1 to 6 were adjusted. They include very high physical security settings to isolate the problems resulting from a trap door, which is contained in the low integrity software when purchased from a vendor, i.e., upon leaving the factory, while eliminating the risks resulting from malicious software introduced into the system from outside. Having only low physical security, an attacker could install new, flawed software, by gaining physical access to the component. To allow for easy detection of an attack event, conditions and triggers have been introduced and the Integrity value has been raised to 100,000.

For test case Reference Monitor Attack 2, “SoftwareType: Reporting” was added to the asset goal, replacing the requirement to use the software “Scare Crow Defense Systems T&M”. There are several instances of this software type. They differ only in their software integrity values. The reason for using the notion of “SoftwareType” instead of the specific software, is to test, whether the game engine recognizes that a specific software, “Scare Crow Defense Systems T&M”, as an instance of the type “Reporting”. Recognizing the software “Scare Crow Defense Systems T&M”, the game engine is expected to correctly determine the integrity of the software as low and therefore generate appropriate attacks. The names of software and the expressions for SDF settings are taken from [Rivermind1 2002].

Test Case 7 is concerned with user trustworthiness and uses a different setting. Wesley is the only user modeled in this scenario. He is working on the single stand-alone component. Wesley’s goal is to modify the single asset, the EW Signatures, which are classified High Integrity EW (HI EW). The asset's secrecy label is CONFIDENTIAL (C). Wesley has the appropriate clearance (C, HI EW), is well trained, but has a low trustworthiness (10). The integrity value of the label HI EW is high (1000), as is its attacker value (800). The secrecy value of the label C is low (100); its attacker value is only 0. The component resides in a walled office, within an office area (CIC) patrolled by a security guard, and protected by a key lock.

Test Cases 8 through 10 model an enhanced scenario, concerned with the notion of wiretaps. The scenarios share the following settings. Wesley and Leonard are the only users, each having his own office, component and asset. Both offices are modeled in a zone with high security settings, including a patrolling security guard, and key lock protection. However, a room separates them. It has very low security settings, and, thus, is accessible by everybody.

Wesley is working in the EW room on a component carrying the asset he has to modify: the EW Signatures classified High Integrity EW (HI EW). The asset's secrecy label is CONFIDENTIAL (C). Leonard is working in the DoC Room on a component carrying the Doctrines (DoC), which he has to modify. The label of the DoC is High Integrity Weapons (HI WP) and C. Both users have the appropriate clearance to access

their specific assets; they are well trained and have a high trustworthiness (100). The Integrity value of label HI EW and HI WP is high (1000), as is its attacker value (800). The Secrecy value of label C is low (100); its attacker value is only 0. Both components are connected over a network. The configuration of the network permits the creation of three new scenarios, as modeled in Test Cases 8, 9, and 10. Each of the test cases, along with its expected and observed results, is described below.

1. Trap Door - Low Integrity Software

The name of the corresponding test-scenario SDF is "Case1.SDF". (Appendix A1a)

This test case models integrity risks caused by low integrity software, as part of the educational concept described in Chapter II, paragraph B, section one. High integrity data is accessed by low integrity software. Other effects, i.e. secrecy, were to be minimal. The test case models one user with a secrecy and integrity clearance appropriate for access to a single high integrity asset that resides on a stand-alone workstation. Both, the software used to access the asset and the component's operating system (OS) software are low integrity. Furthermore, the OS does not enforce a MAC policy.

a. Expected Results

Based on the above configuration, it is expected that the game engine generates attacks aimed at contamination of the asset's integrity, such as Trojan Horse attacks and will report that the asset has been corrupted. The Trojan Horses are expected to be part of malicious software within the low integrity software and OS. The key lock, the guard and the low secrecy attacker motive are expected to prevent attacks on secrecy. The expectation is to experience at least one integrity attack and no attacks on secrecy within 60 days.

b. Experienced Results

As anticipated, the game engine did not generate successful attacks on secrecy, but several successful attacks on the asset's integrity were observed. This behavior fully complies with the expected results. (Run March 4, 2004, using game version March 3; run March 11, 2004, using game version March 10)

Applying changes to the SDF, to make an attack more visible, yielded following results ("Case1.SDF" with changes to above: CM Strong, and IntegrityValue: 100,000):

As anticipated, the game engine did not generate successful attacks on secrecy, but one successful attack on the asset's integrity. This behavior was as expected. (Run March 15, 2004, using game version March 11, run March 30, 2004, using game version March 24).

2. Trap door – High Integrity Software 1

The name of the corresponding SDF is "Case2a.SDF." (Appendix A1b)

This test case models integrity concepts illustrating the use of high integrity software, developed in an environment, practicing rigorous security engineering, as part of the educational concept described in Chapter II, paragraph B, section 2 of this paper. High integrity data is accessed by high integrity software. Other effects, i.e., attacks on secrecy were to be minimal. The test case models one user with a secrecy and integrity clearance appropriate for access to a single high integrity asset that resides on a stand-alone workstation. Both, the software used to access the asset and the component's operating system (OS) are high integrity. However, the OS does not enforce a MAC policy. The component options used in this test case are: OS – GEOS (900) and software - Skunk Cellars RAT (900).

a. Expected Results

Based on the above configuration, it is expected that the game engine does not generate any attacks aimed at the asset's integrity. Therefore no reports on asset corruption are anticipated. The software and the OS are expected to be free of Trojan Horses. Since there is only one high integrity application residing on the component, the lack of a SK is not expected to cause any integrity risks. This expectation, however, is valid only as long as the asset value is below a certain threshold. As known in the security domain, there is no such thing as 100% security, even in a closed environment. In the SDF this is modeled by a maximum attacker value of 800. Above this threshold, an attacker is expected to devote sufficient energy and finances in finding a way to access and corrupt the asset.

The expectation is to not observe any integrity attacks, or any attacks on secrecy within 60 days.

b. Experienced Results

As anticipated, the game engine did not generate successful attacks on secrecy. However, the game engine failed in protecting the asset from modification. This behavior did not meet the expectations. (Run on March 4, 2004, using game version March 3). With CyberCIEGE game version March10 the asset remained protected for a period of 30 days. This complied fully with the expectations. (Run on March 11, 2004, using game version March 10).

Applying changes to the SDF, to make an attack more visible, yields following results ("Case2a.SDF" with changes to: CM Strict, and IntegrityValue: 100,000):

As anticipated, the game engine did not generate successful attacks on secrecy. However, the game engine failed in protecting the asset from modification in the first run. This behavior did not meet the expectations. (Run on March 15, 2004, using game version March11). The second run yielded full compliance. The engine did not generate any successful attack on integrity within 60 days of game play. (Run on March 15, 2004, using game version March 11, run March 30, 2004, using game version March 24).

3. Trap door – High Integrity Software 2

The name of the SDF corresponding to this test-scenario is "Case2b.SDF." (Appendix A1c)

This test case models integrity risks caused by using both, low and high integrity software on the same component. The latter software is developed in an environment, practicing rigorous security engineering. This test case analyzes, whether the game engine is capable of generating behavior expected as part of the educational concept described in Chapter II, paragraph B, section 2. It models the risks faced by high integrity data in the presence of low and high integrity software, yet where a SK is absent. The test case models one user with a secrecy and integrity clearance appropriate for access to a single high integrity asset that resides on a stand-alone

workstation. Both, the software used to access the asset and the component's operating system (OS) are of high integrity. However, the OS does not enforce a MAC policy and the component also incorporates low assurance software designated for other tasks. The component options used in this test case are: OS – GEOS (900); high integrity software - Skunk Cellars RAT (900); low integrity software - Scare Crow Defense Systems T&M (200).

a. Expected Results

Based on the above configuration, it is expected that the game engine generates attacks, aimed at the asset's integrity, and reports about asset corruption. The high integrity, closed environment software and the OS are expected to be free of Trojan Horses. However, malicious code is expected in the low integrity applications. In lack of a SK it is expected that the high integrity asset will be accessed not only by the high integrity application designed and dedicated for modifying the asset, but also by the low integrity applications or a Trojan Horses within.

The expectation is to experience at least one integrity attack and no attacks on secrecy within 60 days.

b. Experienced Results

As anticipated, the game engine did not generate successful attacks on secrecy. The game engine successfully attacked and maliciously modified the asset. This behavior did not meet the expectations. (Run on March 4, 2004, using game version March 3)

However, running the test again with CyberCIEGE game version March10, the game engine performed differently. Again, the game engine did not generate successful attacks on secrecy, which met the expectations. However, contrary to expectations, the game engine did not generate any attack on integrity within 30 game-days.

Applying changes to the SDF, to make an attack more visible, yielded following results ("Case2b.SDF" with changes to: CM Strict, and IntegrityValue: 100,000):

The game engine did not generate successful attacks on secrecy and did not generate successful attacks on integrity within 60 days of game play. This is fully compliant with the expectations. (Run on March 15, 2004, using game version March 11, run March 30, 2004, using game version March 24).

4. MAC Enforcement Mechanism 1

The name of the SDF corresponding to this test-scenario is "Case3a.SDF." (Appendix A1d)

This test case models integrity risks caused by using both, low and high integrity software on the same component. The latter software is developed in an environment, practicing rigorous security engineering. This test case analyzes the game engine's ability to generate behavior expected as part of the educational concept described in Chapter II, paragraph B, section 3. It models the risks faced by high integrity data in the presence of low and high integrity software, while using a security kernel (SK). The test case models one user with a secrecy and integrity clearance appropriate for access to a single high integrity asset that resides on a stand-alone workstation. The software used to access the asset is high integrity, and the component's operating system (OS) is high assurance. However, the component also incorporates low integrity software allocated to other tasks. The OS enforces a MAC policy; thus a SK will mediate the access between the applications and the asset. The component options used in this test case are: OS – Green Shade Core (EAL 7); high integrity software - Skunk Cellars RAT (900); low integrity software - Scare Crow Defense Systems T&M (200).

a. Expected Results

Based on the above configuration, it is expected that the game engine does not generate any successful attacks aimed at the asset's integrity. Therefore no reports on asset corruption are anticipated. The high integrity software and OS are expected to be free of trap doors. The SK is expected to successfully grant the high integrity application designated for operating on the asset access, yet deny access to the low integrity application.

The expectation is to not observe any integrity attacks and no attacks on secrecy within 60 days.

b. Experienced Results

As anticipated, the game engine did not generate successful attacks on secrecy. However, the game engine failed in protecting the asset from modification. This behavior does not meet the expectations. (Run on March 4, 2004, using game version March 3)

Applying changes to the SDF, to make an attack more visible, yields following results ("Case3a.SDF" with changes to above: CM Strict, and IntegrityValue: 10000):

The game engine did not generate successful attacks on secrecy and did not generate successful attacks on integrity within 60 days of game play. This is fully compliant with the expectations. (Run on March 15, 2004, using game version March 11, run March 30, 2004, using game version March 24).

5. MAC Enforcement Mechanism 2

The name of the SDF corresponding to this test-scenario is "Case3b.SDF." (Appendix A1e)

This test case models integrity risks caused by using low integrity software on a component that incorporates a high assurance security kernel (EAL 7). The test case refers to the educational concept described in Chapter II, paragraph B, section 3. It illustrates the risks faced by high integrity data in the presence of low integrity software, while using a SK. The test case models one user with a secrecy and integrity clearance appropriate for access to a single high integrity asset that resides on a stand-alone workstation. The software used to access the asset is of low integrity. The component's OS has a SK which will mediate the access between the application and the asset. The component options used in this test case are: OS – Green Shade Core (MAC EAL 7); low integrity software - Scare Crow Defense Systems T&M (200).

a. Expected Results

Based on the above configuration, it is expected that the game engine generates successful attacks aimed at the asset's integrity, displaying reports that the asset has been corrupted or modified. The SK is expected to perform correctly and to be free of malicious code. However, because the software dedicated to operate on the

asset is of low integrity, malicious code is expected within the application's source code. The SK is expected to correctly grant the low integrity application access to the high integrity asset and thus, the malicious code is anticipated to perform unauthorized and unintended modification.

The expectation is to experience at least one integrity attack and no attacks on secrecy within 60 days.

b. Experienced Results

As anticipated, the game engine did not generate successful attacks on secrecy, but generated a successful attack on the asset's integrity. This behavior fully met the expectations. (Run on March 4, 2004, using game version March 3)

Applying changes to the SDF, to make an attack more visible, yields following results ("Case3b.SDF" with changes to: CM Strict, and IntegrityValue: 100,00):

The game engine did not generate successful attacks on secrecy, or any successful attacks on integrity within 60 days of game play. This did not meet the expectations. (Run on March 15, 2004, using game version March 11).

Using a modified game version, March 24, following results were observed: the game engine did not generate successful attacks on secrecy but did generate one successful attack on integrity within 60 days of game play. This is fully met the expectations (run March 30, 2004, using game version March 24)

6. MAC Enforcement Mechanism 3

The name of the SDF corresponding to this test-scenario is "Case3c.SDF." (Appendix A1f)

This test case models integrity risks caused by using both, low and high integrity software on a component that incorporates a low assurance MAC enforcing mechanism (EAL 4). This test case refers to the educational concepts described in Chapter II, paragraph B, section 3. It illustrates the risks faced by high integrity data in the presence of low integrity software, using a MAC enforcing mechanism. The test case models one user with a secrecy and integrity clearance appropriate for access to a single high integrity asset that resides on a stand-alone workstation. The software

assigned to access the asset is of high integrity. The component's operating system (OS) is of moderate integrity, and an additional application is of low integrity. The OS incorporates a MAC enforcement mechanism of low assurance, which will mediate the access between the applications and the asset. The component options used in this test case are: OS – Trusted Populos Desktop (OS EAL 4, MAC EAL 4); high integrity software – Skunk Cellars RAT (900); low integrity software – WordSmyth.

a. Expected Results

Based on the above configuration, it is expected that the game engine generates several successful attacks aimed at the asset's integrity. Therefore reports on asset corruption are anticipated. The high integrity software is expected to be free of malicious code, such as trap doors. The low assurance MAC enforcing mechanism and the low integrity software, however, are expected to contain malicious code. The MAC enforcing mechanism is expected to successfully grant the high integrity application assigned to operate on the asset access to the high integrity asset. However, since the MAC enforcing mechanism is expected to contain malicious code, it is expected to fail to deny access to the asset by the low integrity application, which has no access permission. Thus, either the malicious code within the low integrity software, or the MAC enforcing mechanism itself are expected to modify the high integrity asset.

The expectation is to observe at least one integrity attack and no secrecy attacks within 60 days.

b. Experienced Results

As anticipated, the game engine did not generate successful attacks on secrecy, but generated a successful attack on the asset's integrity. This behavior fully met the expectations. (Run on March 30, 2004, using game version March 24)

7. Social Engineering Attack

The name of the SDFs corresponding to this test-scenario are "Case4a.SDF." and "Case4b.SDF". (Appendix A1g)

These test cases model integrity risks caused by low user integrity. Case4a models low integrity through a low user trustworthiness value (10), while Case4b uses a low value for the initial user background check (Low). These test cases refer to the educational concepts described in Chapter II, paragraph B, section 4. The test cases

model one user with a secrecy and integrity clearance appropriate for access to a single high integrity asset that resides on a stand-alone workstation. The software designated to access the asset and the component's operating system (OS) are both of high integrity. The component options used in this test case are: OS – Green Shade Core (MAC EAL 7); high integrity software -Skunk Cellars RAT (900).

a. Expected Results

Based on the above configuration, it is expected that the game engine generates successful attacks aimed at the asset's integrity. Therefore several reports about asset corruption are anticipated. The high integrity software and the OS are expected to be free of malicious code, such as trap doors. Although the component is configured safely, it is expected that an attacker will be successful in attacking the asset, by subverting Wesley, the only user, who has a very low integrity.

The expectation is to observe three to five integrity attacks and a maximum of two attacks on secrecy within 10 days.

b. Experienced Results

As anticipated, the game engine did not generate successful attacks on secrecy, but generated a successful attack on the asset's integrity. This behavior fully met the expectations. (Run on March 8, 2004, using game version March 3)

Applying changes to the SDF, to make an attack more visible, yields following results (Changes to: CM Strong, and IntegrityValue: 100,000):

As anticipated, the game engine did not generate successful attacks on secrecy, but generated several successful attacks on the asset's integrity. This behavior fully met the expectations. (Run on March 15, 2004, using game version March 11, run on March 30, 2004, using game version March 24).

8. Network 1

The name of the SDF corresponding to this test-scenario is "CaseNW1a.sdf." (Appendix A1h)

This test case models integrity risks caused by using two highly protected components that reside in different highly protected, but not adjacent zones. These components are attached to a network that spans from zone to zone. The

communication path between the two assets crosses a zone with very low security settings. Each component is directly connected to the same network.

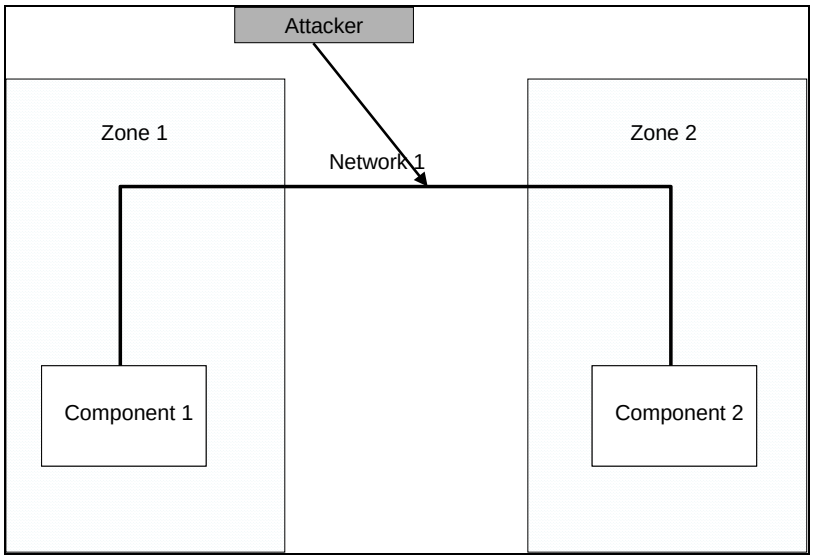

Figure 9. Directly Connected Components

a. Expected Results

Based on the above configuration, it is expected that the game engine generates successful attacks aimed at the asset's integrity. In particular, a wiretap attack is expected, since any attacker would have access to the low security zone, and, hence, would be able to access the network cable.

The expectation is to experience at least one integrity attack and no secrecy attacks within 60 days.

b. Experienced Results

As anticipated, the game engine did not generate successful attacks on secrecy, but generated a successful wiretap attack, compromising the asset's integrity. This behavior fully met the expectations. (Run on March 30, 2004, using game version March 24)

Using CyberCIEGE game version April4, the experienced result was different: the game engine did not generate wiretap attacks, which did not meet the

expectations. However, as expected, the game engine did not generate successful attacks on secrecy, or other attacks on integrity.

9. Network 1 Router

The name of the SDF corresponding to this test-scenario is "CaseNW1a_Router.sdf." (Appendix A1i)

This test case models integrity risks caused by using two highly protected components that reside in different highly protected, but not adjacent zones. These components are attached to a network that spans the zones. The communication path between the two assets crosses a zone with very low security settings. Each component is connected to a router, which is located in one of the protected zones.

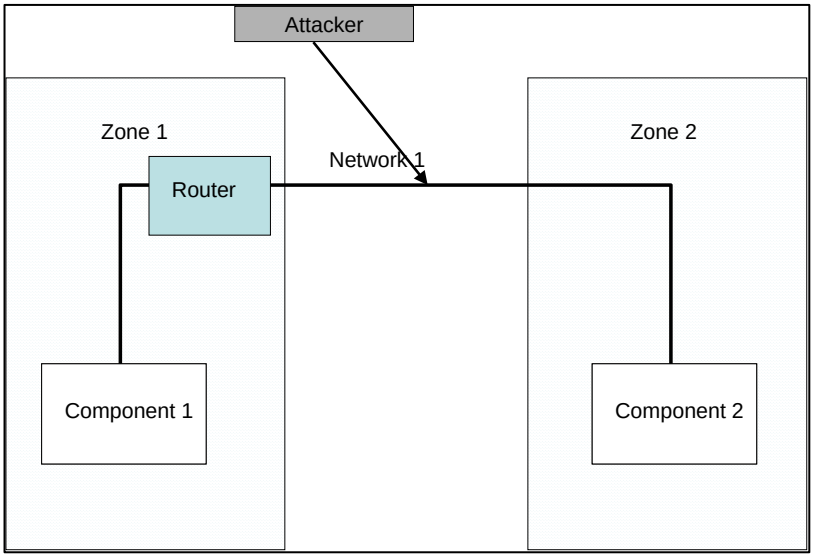

Figure 10. Network With Router

a. Expected Results

Based on the above configuration, it is expected that the game engine generates successful attacks aimed at the asset's integrity. In particular, a wiretap attack is expected, since any attacker has access to the low security zone, hence, is able to access the network cable.

The expectation is to experience at least one integrity attack and no secrecy attacks within 60 days.

b. Experienced Results

As anticipated, the game engine did not generate successful attacks on secrecy, but generated a successful wiretap attack, compromising the asset's integrity. This behavior fully met the expectations. (Run on March 30, 2004, using game version March 24)

Using game version April4, the experienced result was different: the game engine did not generate wiretap attacks, which did not meet the expectations. However, as expected, the game engine did not generate successful attacks on secrecy, or other attacks on integrity.

10. Network 2

The name of the SDF corresponding to this test-scenario is "CaseNW1b.sdf." (Appendix A1j)

This test case models integrity risks caused by using two highly protected components that reside in different highly protected, but not adjacent zones. These components are attached to a network that spans from zone to zone. The communication path between the assets crosses a zone with very low security settings. To ensure security, each component is connected to an encryption device. The encryption devices are directly connected to each other, each residing in one of the protected zones. The communication paths from asset to encryption device run within the highly protected zones. Only the communication path between the encryption devices, carrying encrypted data, runs across the zone with very low security settings

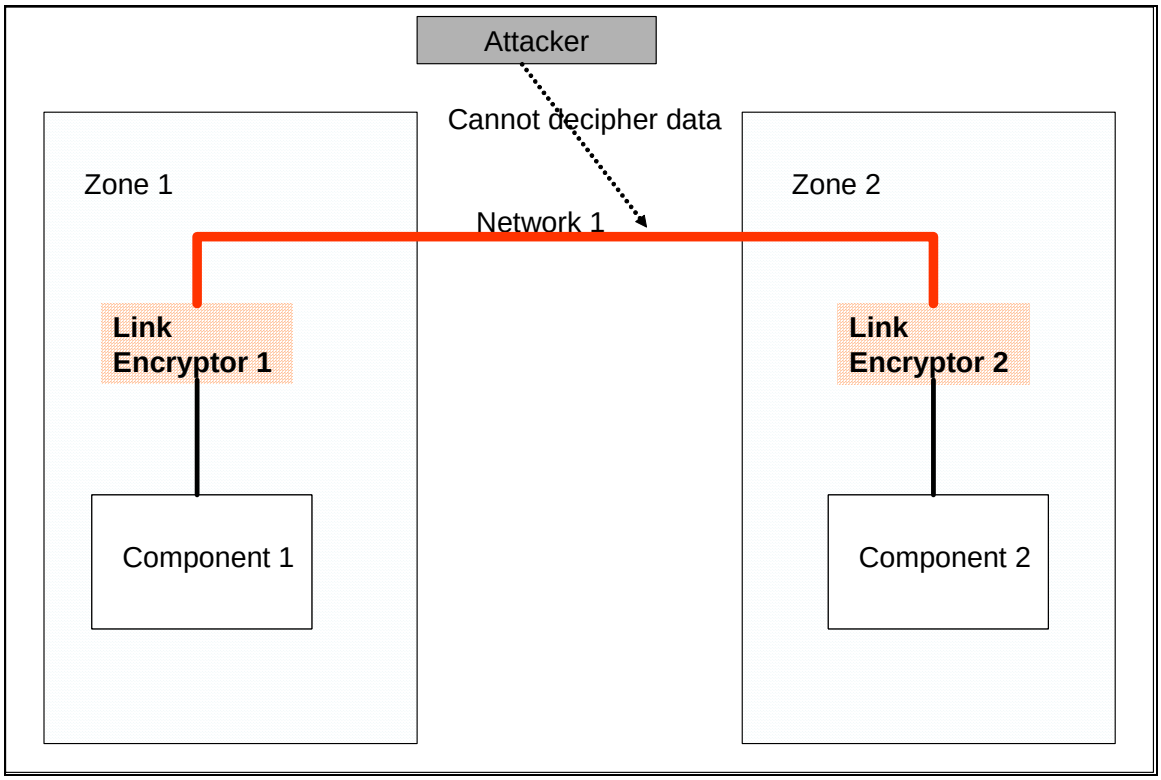

Figure 11. Using A Link Encryptor

a. Expected Results

Based on the above configuration, it is expected that the game engine does not generate any successful attack aimed at the asset's integrity. An attacker could physically insert a wiretap into the unprotected network portion; however, it is assumed that he cannot break the cipher.

The expectation is to experience no integrity attack and no secrecy attacks within 60 days.

b. Experienced Results

As anticipated, the game engine neither generated successful attacks on secrecy, nor successful attacks on integrity. This behavior fully met the expectations. (Run on April 5, 2004, using game version March 24; run on April 6, 2004, using game version April4)

11. Playable Small Game

The name of the SDF corresponding to this test-scenario is "Playable_04_05_03.sdf." (Appendix A1k)

This SDF offers a player the opportunity to learn about the effects of high integrity software on high integrity assets.

The setting is a small military like facility, with two users, Wesley and Leonard. Wesley's goal is to modify the asset Electronic Warfare Signatures (EWS). Leonard needs to modify the asset Doctrines (DoC). The player needs to buy a component for each user, and software of type DEFENSE RAT, so the users can meet their goals. In addition, the player needs to buy physical and procedural security measures, and to assign the assets to the purchased components. To win the game, the player needs to protect the assets from being corrupted for 30 days and to keep the average user productivity above 30%.

One solution that allows the player to win is displayed in Appendix B1k. This solution assumes the player buys the high integrity software *'Skunk Cellars RAT'*, and removes the pre-installed low integrity software *'Molo Defense Systems RAT'*. Furthermore, it is assumed, that the player hires a security guard and buys security measures for the zones EW room and DoC room, such that the physical security value exceeds 430. The procedural security settings also need to be set to high security values, such as strong configuration management (CM), long password length, complex password character set, etc.

One solution reflecting bad player choices, leading to losing the game, is displayed in Appendix A1k. The player is assumed to have bought a security guard and security measures such that the physical security value exceeds 430. However, due to, for example, financial considerations, the pre-installed low integrity software '*Molo Defense Systems RAT*' has been used to allow the users to work on the high integrity assets. In this case, the game engine generates attacks on the assets, resulting in high financial losses, such that the game is lost.

Following table provides an overview of the test cases, their main features and the result of the test run with game engine version May 27th, 2004.

Testcases				GameVersion: May 27			
#	Test #	Name	Description	OS (MAC / OS)	SW	Expectation	Result
1	c1	Case1.SDF	LI SW (LI OS)	Populos V9 Desktop (NA / 40)	WordSmyth Cell Life Viewpoint	modification	ok
2	c2a	Case2a.SDF	HI SW (No MAC / HI OS)	GEOS (NA / 900)	Skunk Cellars RAT (900)	no modification	ok
3	c2b	Case2b.SDF	HI+LI SW (No MAC, HI OS)	GEOS (NA / 900)	Skunk Cellars RAT (900) Scare Crow Defense Systems T&M (200)	modification	ok
4	c3a	Case3a.SDF	SK, **HI** + LI SW (MAC EAL 7 / OS NA)	Green Shade Core (EAL 7 / -)	Skunk Cellars RAT (900) Scare Crow Defense Systems T&M (200)	no modification	ok
5	c3b	Case3b.SDF	SK, **LI** SW	Green Shade Core (EAL 7 / -)	Scare Crow Defense Systems T&M (200)	modification	ok
6	c3c	Case3c.SDF	LA MAC, **HI**+LI SW	Trusted Populos Desktop (EAL 4 /EAL 4)	Skunk Cellars RAT (900) WordSmyth	modification	ok
7	c4a	Case4a.SDF	Social Engineering low user trustworthiness	Green Shade Core (EAL 7 / -)	Skunk Cellars RAT (900)	modification	ok
7b	c4b	Case4b.sdf	Social Engineering low background check	Green Shade Core (EAL 7 / -)	Skunk Cellars RAT (900)	modification	ok
8	nw1a	CaseNW1a.sdf	NW, no protection	GEOS (NA / 900) no LinkEncryptor	Skunk Cellars RAT (900)	modification	NOK
9	nw1aR	CaseNW1a_Router.sdf	NW + Router; no prot.	GEOS (NA / 900) no LinkEncryptor	Skunk Cellars RAT (900)	modification	NOK
10	nw1b	CaseNW1b.sdf	NW + LinkEncr	GEOS (NA / 900) no LinkEncryptor	Skunk Cellars RAT (900)	no modification	NOK
11		Playable_04_05_03.sdf					
bold: has access permission							

Table 7. Test Cases

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS & RECOMMENDATIONS

A. RECOMMENDATIONS

1. CyberCIEGE Game Engine Improvements

In the course of SDF development, and testing of the CyberCIEGE game engine, some game-play issues were experienced, which should be addressed in future updates of the game engine. The recommendations made in this section are based on tests using the CyberCIEGE game engine version May 27th, 2004.

a. Mandatory Access Control Enforcing Mechanisms

The CyberCIEGE game provides components with different Operating Systems (OS). Some of these OSes provide mandatory access control (MAC) enforcing mechanisms of different assurance levels. While the OSes with high assurance MAC enforcing mechanisms correctly protect high integrity data from modification on stand-alone components, the game engine generates successful attacks on the high integrity data, if these components are connected to a network with other components that have low assurance OSes. This behavior does not simulate real world behavior, where a high assurance MAC enforcing mechanism protects high integrity data independent of the nature of the other components connected to the same network. This game engine behavior needs to be corrected.

b. Wiretap Attacks

The CyberCIEGE game provides the means to connect components to networks. If the network architecture is such, that the communication path between components holding high integrity assets crosses a zone with low security settings, the game engine is supposed to generate wire tap attacks. The game engine version May 27th, 2004, however, does not generate these attacks. This issue needs to be addressed in future game engine improvements, to better comply with real world behavior.

c. Software

The CyberCIEGE game allows the player to buy software for components, to accommodate user needs. This software, however, is bound to the component and is lost, if the player decides to sell the component. If a specific component needs to be replaced during game-play, the player is forced to pay for the

software a second time. To generate more realistic behavior, it is suggested to make the software a separate entity, possibly represented by an icon, such that the player can re-install the software on the component of his choice.

d. Graphics

The SDFs produced for this thesis model the combat information center (CIC) on board a modern battle ship. The only graphics-file available for the CyberCIEGE game, at completion of the thesis, that represents a military-like facility, is based on an office showing a shore facility. To better model the CIC environment of a battle ship, a modified graphics-file is recommended, which instead of displaying brick walls, and grass at the outer perimeter, models the interior of a battle ship.

e. User Sensitivity

The CyberCIEGE game, among other parameters, models user happiness. The value of the user happiness is influenced by the user's sensitivity to security measures. For example, user happiness suffers in the presence of measures like patrolling guards, cameras, iris scanners, etc. While this sensitivity is important to teach the player that, sometimes, a careful balance is needed between security measures and a suitable working environment, the game engine needs to be adopted to distinguish between civilian and military users. Military computer users are used to work in high security zones and are less negatively affected by strict security measures, than their civilian counterparts.

2. Future Work

The testing approach used for this project is the Verification Validation and Accreditation (VV&A) [DMSO2 2003], more specifically, the face validation process as part of VV&A. The testing concentrated on verifying, that the game engine properly simulates real world behavior.

The next interesting step is to generate tests, to evaluate the effectiveness of the CyberCIEGE game in educating and training players. Results from such testing might lead to further improvements concerning the game engine or future SDFs.

The first SDFs created for the CyberCIEGE project concentrated on specific aspects of computer security principles. Future SDFs may build on the basics and concentrate on more complex scenarios.

B. CONCLUSIONS

The objective of this thesis was to contribute to ongoing research concerning the project CyberCIEGE, conducted at the Naval Postgraduate School. “The purpose of the CyberCIEGE project is to create an Information Assurance (IA) teaching/learning laboratory.”[Irvine1 2003]

This thesis asked if a Scenario Definition File (SDF) for the CyberCIEGE game could be developed, to educate and train players in Information Assurance on matters related to information integrity in a networking environment. The primary educational concern was the protection of stored data. Two SDFs were developed for teaching purposes. The SDFs demonstrate that, indeed, it is possible to create SDFs for the CyberCIEGE game engine to teach specifically about integrity issues. Future work is needed to evaluate the degree of effectiveness of this educational tool.

A secondary goal was to test whether the CyberCIEGE game engine properly simulates real world behavior. Several SDFs were developed to demonstrate the game engine’s ability to simulate real world behavior for specific, isolated educational goals. This thesis evaluated many results and proposed changes to the game engine, which improved its behavior. The CyberCIEGE project is still in the development phase. Thus, the game engine is still being updated. Although not all of the experienced flaws were corrected at the time of the completion of this thesis, they will be considered for future game engine updates.

As the number of computer users continues to grow, and attacks on assets stored on computer devices increase, effective education and training of computer users and policy makers in Information Assurance becomes more important. The CyberCIEGE game can be used to convey requisite facts and to generate tacit understanding of general computer security concepts to a broad audience, and thus, encourage computer users to implement security principles in their daily lives.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A: SOFTWARE DEVELOPMENT FILES (SDF)

This Appendix includes the SDF's of the main test cases described in this paper, the playable SDF and a version with possible player choices that lead to winning the game, and a version with bad player choices that lead to loosing the game.

The Appendix is accessible via following link:
http://library.nps.navy.mil/uhtbin/cgisirsi/Tue+Aug+31+12:05:24+PDT+2004/SIRSI/0/520/04Sep_Fielk_Appendix.doc

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B: WORKSPACE FILES

The workspace file is specified in the SDF and determines the position of the user workspaces. The position is a coordinate consisting of two numbers. The first number describes the latitude, and the second number describes the longitude of the position at which the workspace is generated within the scenario. The first letter, N, S, E, or W, indicates in what direction the workspace faces. The last letter, A, or I indicate whether a workspace will be active (A) or inactive (I). If the workspace is inactive, it is not drawn in the graphical representation of the SDF.

The Appendix is accessible via following link:
http://library.nps.navy.mil/uhtbin/cgisirsi/Tue+Aug+31+12:05:24+PDT+2004/SIRSI/0/520/04Sep_Fielk_Appendix.doc

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[Anderson 2002] Anderson, E.A., *A Demonstration of the Subversion Threat: Facing a Critical Responsibility in the Defense of Cyberspace*, Department of Computer Science, 2002, Naval Postgraduate School: Monterey, CA. URL: http://library.nps.navy.mil/uhtbin/cgisirsi/Mon+Aug+30+11:14:23+PDT+2004/SIRSI/0/520/02Mar_AndersonE.pdf

[Blohm + Voss 2004] Blohm + Voss GmbH, 2004, *Type ship F124 "Sachsen"*, retrieved on April 7, 2004, Blohm + VossGmbH webpage: http://212.72.173.53/media/9f1909274389f70c7503b5614de8c549.pdf

[Brinkley 1995] Brinkley, D.L. and Schell, R.R. (1995). *Concepts and Terminology for Computer Security*. Essay 2 of: Information Security: An Integrated Collection of Essays; Edited by Marshall D. Abrams, Sushil Jajodia, Harold J. Podell, IEEE Computer Society Press, Los Alamitos, California USA ISBN: 0-8186-3662-9, LoC CIP: 94-20899, DDN: QA76.9.A25I5415. Retrieved November 21, 2002 from World Wide Web: http://www.acsac.org/secshelf/book001/02.pdf

[CC 2004] CC 2004**,** *Common Criteria for IT Security Evaluation***,** created November 12, 2002, Last updated July 26, 2004; retrieved July 27, 2004 from the National Institute of Standards and Technology (NIST) home page: http://csrc.nist.gov/cc/index.html

[Clark 1987] Clark, D.D., Wilson, D.R., *A Comparison of Commercial and Military Computer Security Policies*, Proceedings of the 1987 IEEE Symposium on Security and Privacy (Cat. No. 87CH2416-6), pp. 184-94, IEEE Computer Society Press, Washington, DC, 1987

[CNSS No. 4009] CNSS Instruction No. 4009**,** *National Information Assurance (IA) Glossary*, Revised May 2003; retrieved April 8, 2004 from: http://www.nstissc.gov/Assets/pdf/4009.pdf

[DOD 1997] DoD 5000.59-M: *DoD Modeling and Simulation (M&S) Glossary*, December 1997

[DON 1999] Department of the Navy (DON) Information Security Program (ISP) Regulation, *SECNAV Instruction 5510.36 Chapter 4, Classification Management*; retrieved from: http://neds.nebt.daps.mil/551036.htm, last revised on: 17 March 1999

[DMSO1 2001] DMSO, Subtopic: *Informal V&V Techniques,* Subtopic to Verification, Validation, and Accreditation (VV&A) Recommended Practices Guide (RPG); Retrieved on March 12, 2004 from: http://vva.dmso.mil/Mini_Elabs/VVtech-informal.htm#inf3, last revised on 8/15/01

[DMSO2 2003] DMSO, *Verification, Validation, and Accreditation (VV&A) Recommended Practices Guide (RPG*): Retrieved January 24, 2003, from: http://vva.dmso.mil/Special_Topics/HBR-Validation/default.htm>

[Ferraiolo 1992] Ferraiolo, D., Kuhn, R., *Role-Based Access Control*, Reprinted from *Proceedings of 15th National Computer Security Conference, 1992*

[Hatcher 2001] Hatcher, Thurston, 2001, *Survey: Costs of computer security breaches soar*, posted on the CNN webpage on March 12, 2001, retrieved on March 31, 2004, from:

http://www.cnn.com/2001/TECH/internet/03/12/csi.fbi.hacking.report/

[Irvine1 2003] Irvine, C. & Thompson, M. *Teaching Objectives of a Simulation Game for Computer Security.* Retrieved August 14, 2003 from NPS Intranet: http://cisr.nps.navy.mil/downloads/project_simsecure2.pdf

[Irvine2 2002] Irvine, C. & Levin, T.E.; *A Cautionary Note Regarding The Data Integrity Capacity Of Certain Secure Systems;* Integrity, Internal Control and Security in Information Systems, ISBN 1-4020-7005-5, Brussels Belgium, 2002

[Irvine3 200] Irvine, C. Class note for CS 4600, Secure Systems, NPS, Monterey 2004

[Irvine4 2004] Irvine, C.; Teaching in CS 4600, Secure Systems, at NPS, Monterey

[Johns 2004] Johns, K., 2004, *Toward Managing & Automating CyberCIEGE Scenario Definition File Creation*, NPS, Monterey, 2004

[Karger 1974] Karger, P. A., and Schell, R., *Multics Security Evaluation: Vulnerability Analysis*, ESD-TR-74-193 Vol. II, ESD/AFSC, Hanscom AFB, Bedford, MA 01731 (June 1974)

[Minihan 1997] Minihan, K., Lt. Gen., 1997, as quoted in *NSA Chief: Attacks on Military Computers Rise*, October 24, 1997; retrieved from wired news webpage on April 8, 2004:

http://www.wired.com/news/politics/0,1283,7963,00.html

[NCSC 1988] NCSC-TG-004-VERSION-1, *Glossary of Computer Security Terms*, National Computer Security Center, Fort George G. Meade, October, 21, 1988

[Nua Internet Surveys 2004] Nua Internet Surveys, Jupitermedia corporation; Retrieved on April 12, 2004 from: http://www.nua.ie/surveys/how_many_online/

[Richter 2003] Richter, J.; An Analysis Of Synergies Of It-Applications And Knowledge Management Strategies With Regard To Organizational Change, NPS, Monterey, 2003

[Rivermind1 2002] Rivermind, Inc. & Naval Postgraduate School Center for Information Systems Security Studies and Research (2002). *CyberCIEGE: Scenario Format Template*

[Rivermind2 2002] Rivermind, Inc. & Naval Postgraduate School Center for Information Systems Security Studies and Research (2002). *CyberCIEGE: Game and Encyclopedia*

[Safire 2004] Safire, William, *The Farewell Dossier*, section A; page 21; column 6; editorial desk; The New York Times, Late Edition, February 2, 2004

[Sandhu 1995] Sandhu, R.S. and Jajodia, S., *Integrity Mechanisms in Database Management Systems*, Essay 27 of: Information Security: An Integrated Collection of Essays; Edited by Marshall D. Abrams, Sushil Jajodia, Harold J. Podell, IEEE Computer Society Press, Los Alamitos, California USA ISBN: 0-8186-3662-9, LoC CIP: 94-20899, DDN: QA76.9.A25I5415. Retrieved November 21, 2002 from World Wide Web: http://www.acsac.org/secshelf/book001/027.pdf

[Schell 1995] Schell, R.R. and Brinkley, D.L. (1995), *Evaluation Criteria for Trusted Systems*. Essay 6 of: Information Security: An Integrated Collection of Essays; Edited by Marshall D. Abrams, Sushil Jajodia, Harold J. Podell, IEEE Computer Society Press, Los Alamitos, California USA ISBN: 0-8186-3662-9, LoC CIP: 94-20899, DDN: QA76.9.A25I5415. Retrieved November 21, 2002 from World Wide Web: http://www.acsac.org/secshelf/book001/06.pdf

[Sullivan 2002] Sullivan, A., 2002, *Experts Easily Crack Government Computers*, posted on the ABC webpage on August 16, 2002, retrieved on April 8, 2004, from:

http://abcnews.go.com/sections/scitech/DailyNews/militaryPChack020816_wire.html

[The Business Journal 2003] The Business Journal, October 24, 2003 *Survey: U.S. tops 150 million Internet users*, retrieved on March 31, 2004 from The Business Journal webpage: http://www.bizjournals.com/portland/stories/2003/10/20/daily51.html

[Thompson 2004] Thompson, M. Discussion on April 20, 2004

[Usher 2003] Usher, A., *Towards a Taxonomy of Information Assurance*, Updated October 3, 2003; Retrieved from the World Wide Web on November 10, 2003:

http://www.sharp-ideas.net/ia/information_assurance.htm

[WASHINGTON (AP) 1999] WASHINGTON (AP), 3.22.1999, as posted by *(ISN) Military computers vulnerable*, Mar 23, 1999 on ISN web page; retrieved on April 8, 2004, from:

http://www.landfield.com/isn/mail-archive/1999/Mar/0063.html

[WEB.DE 2004] WEB.DE, FreeMail, WEB.DE Viruswarnung, retrieved March 8, 2004, URL:

https://freemailng1104.web.de/online/frame.htm?si=9EOO.1b0zLw.1HOvsS.1t**&v=1

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Ken Allen
   Rivermind
   Mountain View, CA

4. George Bieber
   OSD
   Washington, DC

5. RADM Joseph Burns
   Fort George Meade, MD

6. Bill Chinn
   Rivermind
   Mountain View, CA

7. Deborah Cooper
   DC Associates, LLC
   Roslyn, VA

8. CDR Daniel L. Currie
   PMW 161
   San Diego, CA

9. Louise Davidson
   National Geospacial Agency
   Reston, VA

10. LCDR James Downey
    NAVSEA
    Washington, DC

11. Scott Gallardo
    Rivermind
    Mountain View, CA

12. Richard Hale
    DISA
    Falls Church, VA

13. LCDR Scott D. Heller
    SPAWAR
    San Diego, CA

14. Wiley Jones
    OSD
    Washington, DC

15. Hun Kim
    Department of Homeland Security
    Washington, DC

16. Russell Jones
    N641
    Arlington, VA

17. David Ladd
    Microsoft Corporation
    Redmond, WA

18. Frank Larry
    Department of Homeland Security
    Washington, DC

19. Dr. Carl Landwehr
    National Science Foundation
    Arlington, VA

20. Steve LaFountain
    NSA
    Fort Meade, MD

21. Dr. Greg Larson
    IDA
    Alexandria, VA

22. Penny Lehtola
    NSA
    Fort Meade, MD

23. Gilman Louie
In-Q-Tel
Menlo Park, CA

24. Ernest Lucier
Federal Aviation Administration
Washington, DC

25. CAPT Sheila McCoy
Headquarters U.S. Navy
Arlington, VA

26. Dr. Diana Gant
National Science Foundation
Arlington, VA

27. Dr. Vic Maconachy
NSA
Fort Meade, MD

28. Doug Maughan
Department of Homeland Security
Washington, DC

29. John Mildner
SPAWAR
Charleston, SC

30. Dr. John Monastra
Aerospace Corporation
Chantilly, VA

31. Brian Morgan
Rivermind
Mountain View, CA

32. Marshall Potter
Federal Aviation Administration
Washington, DC

33. Dr. Roger R. Schell
Aesec
Pacific Grove, CA

34. Keith Schwalm
Good Harbor Consulting, LLC
Washginton, DC

35. Dr. Ralph Wachter
ONR
Arlington, VA

36. David Wirth
N641
Arlington, VA

37. Daniel Wolf
NSA
Fort Meade, MD

38. CAPT Robert Zellmann
CNO Staff N614
Arlington, VA

39. Albert Wong
Naval Postgraduate School
Monterey, CA

40. Michael F. Thompson
Naval Postgraduate School
Monterey, CA

41. Dr. Cynthia E. Irvine
Naval Postgraduate School
Monterey, CA

42. Paul C. Clark
Naval Postgraduate School
Monterey, CA

43. Ann E. Rideout
SPAWAR
Charleston, SC

44. Klaus Fielk
Naval Postgraduate School
Monterey, CA